Tutorial: Thermal conductivity from EMD

From GPUMD
Jump to navigationJump to search

Contents

Introduction

- In this example, we use the EMD (Green-Kubo) method to calculate the lattice thermal conductivity of graphene at 300 K and zero pressure.
- All the input and output files can be found here (https://github.com/brucefan1983/GPUMD/tree/master/examples/gpumd/thermal_transport_emd).

Preparing the Inputs

The xyz.in file

- The xyz.in file used is the same as in Tutorial: Density of states.
- Note that the thickness of the graphene sheet is set to 3.35 A according to the convention in the literature. This thickness is needed to calculate an effective 3D thermal conductivity for a 2D material.

The `run.in` file

- The `run.in` input file is given below:

```
potential    potentials/tersoff/Graphene_Lindsay_2010_modified.txt
velocity     300

# equilibration
ensemble     npt_ber 300 300 0.01 0 0 0 0.0005
time_step    1
dump_thermo  1000
run          1000000

# production
ensemble     nve
compute_hac  20 50000 10
run          10000000
```

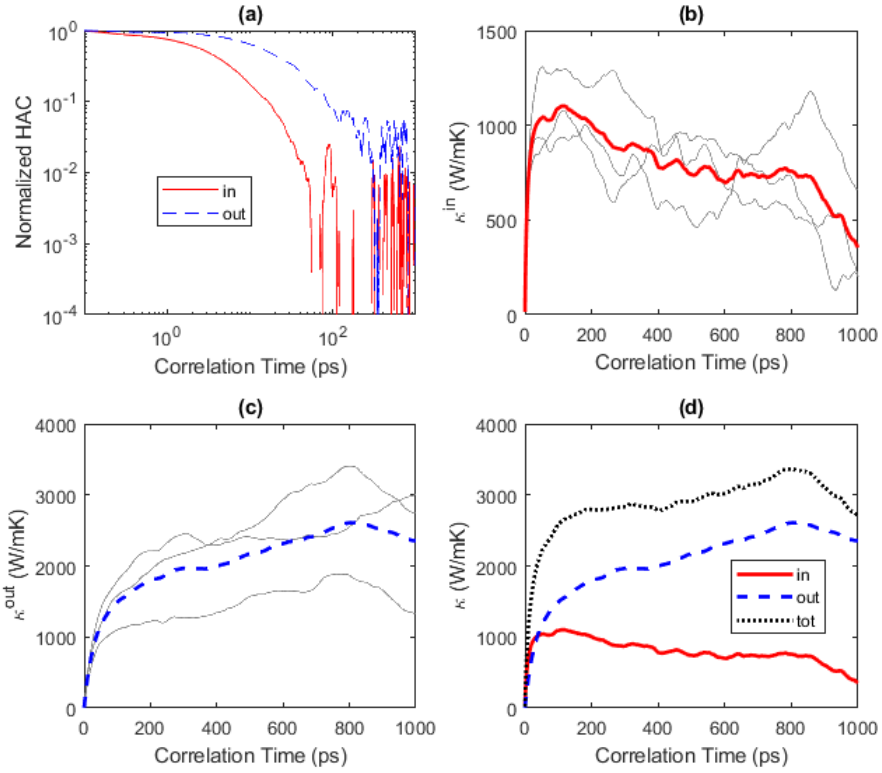
- The first line of command with the potential keyword tells that the potential to be used is specified in the file Graphene_Lindsay_2010_modified.txt (https://github.com/brucefan1983/GPUMD/blob/master/poten

tials/tersoff/Graphene_Lindsay_2010_modified.txt).

- The second line of the command with the velocity keyword tells that the velocities will be initialized with a temperature of 300 K.

- There are two runs.
  - The first run serves as the equilibration stage, where the NPT ensemble (the Berendsen barostat) is used. The temperature is 300 K and the pressures are zero in all the directions. The coupling constants are 0.01 (dimensionless) and 0.0005 (in the natural unit system adopted by GPUMD) for the thermostat and the barostat, respectively. The time step for integration is 1 fs. There are 10^6 steps (1 ns) for this run and the thermodynamic quantities will be output every 1000 steps.
  - The second run is for production. Here, the NVE ensemble is used. The line with the compute_hac keyword means that heat currents will be recorded every 20 steps (20 fs), 50000 HAC data (the maximum correlation time is then about 1 ns) will be calculated, and the HAC are averaged for every 10 data points before written out. The production time is 10 ns (10^7 steps), which is 10 times as long as the maximum correlation time. This is a reasonable choice.

Results and discussion



Thermal conductivity results for pristine graphene at 300 K from EMD simulations. (a) Normalized HAC as a function of correlation time for the in-plane and out-of-plane components. (b) Individual (thin lines) and averaged (thick line) RTC as a function of correlation time for the in-plane component. (c) Individual (thin lines) and averaged (thick line) RTC as a function of correlation time for the out-of-plane component. (d) Averaged RTC as a function of correlation time for various components.

Computation time

- The above figure shows the results from three independent runs, which took about two hours in total using a Tesla K40 card.

Results

- As the system is essentially isotropic in the planar directions, we only consider a scalar thermal conductivity $\kappa = \left(\kappa_{xx} + \kappa_{yy}\right)/2$ for the 2D system. However, we consider the **in-out decomposition** as introduced in [Fan 2017].
- From (a), we can see that the in-plane component and the out-of-plane component of the HAC have different time scales. The latter decays much more slowly.
- Panel (b) shows the individual and averaged RTCs for the in-plane component $\kappa^{\rm in}\left(t\right)$. The averaged RTC converges to about 1000 W/mK at around 100 ps.
- Panel (c) shows the individual and averaged RTCs for the out-of-plane component $\kappa^{\rm out}\left(t\right)$, and the convergence property is not very clear here. This is because the out-of-plane component converges very slowly and three independent simulations (each with 10 ns) are not enough to give accurate results.
- Summing up $\kappa^{\rm in}\left(t\right)$ and $\kappa^{\rm out}\left(t\right)$, we get $\kappa^{\rm tot}\left(t\right)$, as shown in panel (d).

Discussion

- Accurately calculating thermal conductivity of graphene using the EMD method can be a very time consuming task. The results we presented here are from three independent simulations with a total production time of 30 ns. It can been seen that the HAC data already become very noisy when the correlation time is 100 ps. To obtain accurate results, one needs to do many independent simulations. Much more accurate data were presented in Fig. 2 of [Fan 2017]. Here are the simulation parameters used in [Fan 2017] which differ from those used in this example:
  - The simulation cell size used in [Fan 2017] is larger, which is about 25 nm x 25 nm (24000 atoms), instead of 15 nm x 15 nm (8640 atoms) here.
  - The maximum correlation time used in [Fan 2017] is larger, which is 10 ns, instead of 1 ns here.
  - The production time used in [Fan 2017] for one independent simulation is larger, which is 50 ns, instead of 10 ns here.
  - There were 100 independent simulations in [Fan 2017], instead of 3 here. Therefore, the total production time used in [Fan 2017] is 5000 ns.
  - Each independent simulation in [Fan 2017] took about 10 GPU hours (using Tesla K40) and about 1000 GPU hours were used to obtain the results shown in Fig. 2 of [Fan 2017].
  - We see that the EMD method can be very time consuming. A more efficient method of computing thermal conductivity is the HNEMD method, which is discussed in another tutorial: Tutorial: Thermal transport from NEMD and HNEMD.

References

- [Fan 2017] Zheyong Fan, Luiz Felipe C. Pereira, Petri Hirvonen, Mikko M. Ervasti, Ken R. Elder, Davide Donadio, Tapio Ala-Nissila, and Ari Harju, *Thermal conductivity decomposition in two-dimensional materials: Application to graphene* (https://doi.org/10.1103/PhysRevB.95.144309), Phys. Rev. B **95**, 144309 (2017).