Temperature Control in Molecular Dynamics

SI2530 – Computational Physics

Ludvig Doeser

October 27, 2020

Abstract

In molecular dynamics (MD) simulations one has to choose between different thermostats to achieve a fluctuating instantaneous temperature around a constant reference temperature. Two such are the Berendsen thermostat (BT) and the Nosé-Hoover thermostat (NHT). In this paper, the previously known results for these two thermostats in terms of stability and fluctuations are reproduced. In particular, it is shown how the time parameters, a low τ-value in BT and a high ω-value in NHT, increase the thermostats' performance in terms of faster responses and lower fluctuations. During changes in reference temperature, however, a too high ω leads to oscillations, so in this respect, the BT is more successful. On the contrary, the BT fails to accurately generate energy fluctuations corresponding to the thermodynamics of a canonical ensemble while the NHT succeeds; it is shown that NHT generates a heat capacity per particle of $C_V = 2.67k_B$ at $T = 85$K, which compares very well to the experimental value of $C_{V,\text{exp}} = 2.78k_B$. To switch between the thermostats, using mainly NHT but BT during changes in T_{ref}, would therefore enhance the overall performance.

1 Introduction

In an NVT-ensemble the number of particles, the volume, and the temperature ought to be constant. To correctly represent a canonical ensemble, however, the energy, and in turn, the temperature, need to fluctuate. In an experimental set-up where one connects one's system to a heat bath, this is also what will happen. In fact, statistical mechanics tells us there is no ambiguity here; one can have an instantaneous temperature fluctuating around a constant value and thus still be in the NVT-ensemble [1].

In a molecular dynamics (MD) simulation one can use different thermostats to achieve a constant temperature. Unfortunately, not all thermostats allow for the fluctuations in energy to obey the relation derived in a canonical ensemble (see Eq. (16)).

The equation of motion in a system of particles is

$$\frac{d\bar{p}_i}{dt} = -\nabla U_i - \xi \bar{p}_i, \tag{1}$$

where $\bar{p}_i$ is the momentum of particle i, that is being exposed to a force $F = -\nabla U$ and some fluctuations depending on ξ. In reduced units (see more in Section 2 and Appendix 5.1), e.g. $m_{atom} \mapsto 1$, this can be expressed as

$$\frac{d\bar{v}_i}{dt} = F_i - \xi \bar{v}_i. \tag{2}$$

Two well-used thermostats that differ in how ξ is chosen are *Berendsen* and *Nosé-Hoover*. As shown in [1, 2] the Berendsen thermostat (BT) does not manage to produce the fluctuations needed for a canonical ensemble. On the contrary, the Nosé-Hoover thermostat (NHT) does succeed, and one therefore needs to use the NHT to correctly simulate a canonical ensemble. Nonetheless, when it comes to stability and robustness during changes in reference temperature, the Berendsen implementation is superior [3, 4].

The main aim of this report is to reproduce these results and verify the claims previously made in, e.g., [1, 2, 5].

The article is organized so that the NHT and the BT, together with the methods used, are presented in Section 2. The section ends with listing the investigations to be made, and Section 3 presents the results in terms of differences between the NHT and the BT, as well as how the time parameter in each thermostat affects the performance. Lastly, Section 4 summarizes, discusses, and concludes the results presented in the previous section.

2 Method

A simple molecular dynamics program implemented in FORTRAN has been used to simulate a Lennard-Jones (LJ) fluid of $N = 500$ argon atoms in the NVT-ensemble. The LJ-potential is defined as

$$U_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{3}$$

where ϵ and σ are two independent parameters. These are used to do all calculations in reduced

units, where

$$\begin{aligned} r &\mapsto r^* = r\sigma^{-1} \\ E &\mapsto E^* = E\epsilon^{-1}. \end{aligned} \tag{4}$$

These, together with the mass M of a single argon atom and Boltzmann's constant k_B, can be used to derive all other quantities' non-dimensional expressions, which are listed in Appendix 5.1. For convenience, we will drop the $*$-notation and only use reduced units from hereon, unless stated otherwise.

To investigate the two aforementioned thermostats' performance two different temperatures of argon were analyzed; $T = 0.29216$, corresponding to a solid form at 35K, and $T = 0.70952$, corresponding to liquid form at 85K.

The system was either initialized by generating velocities at these temperatures, or by using the output coordinates and velocities from a previous simulation with the corresponding thermostat. The generation was made by randomly assigning velocities, calculating the temperature T this gives rise to, and then scaling by $\sqrt{T_{\text{ref}}/T}$ (see Eq. (7)).

Thereafter, the equation of motion Eq. (2) determined the evolution with time depending on the choice of the thermostat type. Lastly, the data analysis was performed by extracting relevant data and using MATLAB for plotting.

2.1 Berendsen Thermostat, BT

To achieve a constant temperature one can simply use a scaling factor λ on the velocity after each iteration as

$$\bar{v}_i \mapsto \lambda \bar{v}_i. \tag{5}$$

Using $T = 2\frac{E_k}{N_{df}} = \frac{1}{2}\sum_{i=1}^{N} 2\frac{v_i^2}{N_{df}}$, where N_{df} is the total number of degrees of freedom and taking $\Delta T = T(t;\lambda v) - T(t;v)$ at time t, one get the relation

$$\Delta T = \left(\lambda^2 - 1\right) T. \tag{6}$$

The simplest scaling

$$\lambda = \sqrt{\frac{T_{\text{ref}}}{T}} \tag{7}$$

would result in a perfectly constant temperature $\Delta T = 0$. However, to represent reality the temperature ought to be able to fluctuate, and this scaling is therefore not desired. Instead, in the BT we construct a thermostat which permits temperature fluctuations through

$$\frac{dT(t)}{dt} = \frac{1}{\tau}(T_{\text{ref}} - T(t)), \tag{8}$$

where τ is a time parameter which determines the speed of convergence towards the reference temperature T_{ref}. This can be written discretely for each time step as $\Delta T = \frac{\delta t}{\tau}(T_{ref} - T(t))$, from which one together with Eq.(6) gets

$$\lambda^2 = 1 + \frac{\delta t}{\tau}\left\{\frac{T_{\text{ref}}}{T\left(t - \frac{\delta t}{2}\right)} - 1\right\}, \tag{9}$$

where $T\left(t - \frac{\delta t}{2}\right)$ comes from the use of the leap-frog integrator.

An alternative, equivalent, formulation of how to update the system, instead of $\bar{v}_i \mapsto \lambda \bar{v}_i$, is to directly iterate in accordance with Eq. (2). By differentiating $\bar{v} = \lambda \bar{v}'$ wrt time, where $\bar{v}'$ is the velocity before the scaling, and comparing with Eq. (2), one can identify ξ as

$$\xi(t) = -\lim_{\Delta t \to 0} \frac{\lambda(t;\Delta t) - 1}{\Delta t} = -\left.\frac{\partial \lambda(t;\Delta t)}{\partial(\Delta t)}\right|_{\mathcal{C}}, \tag{10}$$

where $\mathcal{C}$ is the constraint that $\Delta t = 0$ [2]. Doing a binomial approximation $(1+x)^a \approx 1 + ax$ on Eq. (9), which is justified by the temperature fluctuation being sufficiently small and $\delta t \ll \tau$, and then using Eq. (10), one receives

$$\xi = -\frac{1}{2\tau}\left(\frac{T_{\text{ref}}}{T} - 1\right) \tag{11}$$

in reduced units. In fact, as shown in [2], Eq. (11) is achieved even if not doing the approximation.

2.2 Nosé-Hoover Thermostat, NHT

Instead of using ξ given by Eq. (11), we now let ξ change with time (see [2]) according to

$$\frac{d}{dt}\xi = -\omega^2 N_{\text{atoms}}\left(\frac{T_{\text{ref}}}{T} - 1\right), \tag{12}$$

with $\xi(0) = 0$ and where ω here plays the same role as τ. Note that N_{atoms} only scales the derivative to let smaller values of ω have a larger influence.

2.3 Problem formulation

To investigate how the two thermostats differ, and what effect the two time parameters τ and ω have, the following areas will be analyzed:

- The temperature stability and fluctuations.
- The responses of the system to small sudden changes in the temperature of the heat bath.
- The obedience of the velocities with a Maxwellian distribution.
- The heat capacity from the canonical ensemble equation based on energy fluctuations, and the heat capacity obtained from temperature fluctuations in a microcanonical (NVE).

In the coming section, the performance of the two thermostats will be presented. As the number of iteration steps varied, this will be presented briefly before the results in the coming section.

3 Results & Analysis

3.1 Stability and Fluctuations in T

Different values of τ and ω and their influence were investigated by performing simulations of 10000 iterations at $T = 0.29216$ with the Berendsen thermostat (BT) and the Nosé-Hoover thermostat (NHT) respectively. The results are presented in Fig. 1 and 2. First of all, we note an initial dip in all cases, which is a result of the velocities being randomly generated. One could avoid these by first letting the thermostat work for a while and then use the output-coordinates; this approach, however, displays better how the time parameter affects the initial performance.

For the BT in Fig. 1 it can be seen that a low value of τ, all the while non-zero, results in a shorter rise time, i.e. the time required for the response to reach within 5% of its final value T_{ref}. While $\tau = 0.05$ rises in around $t \approx 0.4$ it takes $t > 10$ for $\tau = 5$.

In Fig. 2 a higher value of ω in the NHT leads to a shorter rise time. It can be seen that $\omega = 0.5$ gives not only an initial dip but also quite a large overshoot, before settling around $t = 0.6$, although with considerable fluctuations. A value of $\omega = 10$ leads to an immediate settling, similar to what a very small $\tau \ll 0.05$ would lead to in the BT.

These results are to be expected as a high value of τ in the BT leads to a small ξ, so it will take a longer time before the second term of Eq. 2 makes a difference. In the NHT, however, a low ω will lead to a tiny update of ξ during each iteration, resulting in a similar effect.

To further investigate how these parameters affect the temperature fluctuation σ_T around T_{ref} we use

$$\sigma_T = \frac{1}{N} \sum_i (T_i - T_{\text{ref}})^2 \,. \tag{13}$$

For several values of τ and ω a simulation of 60000 iterations was performed and the data was divided into subsets of 5000 steps each. Disregard-

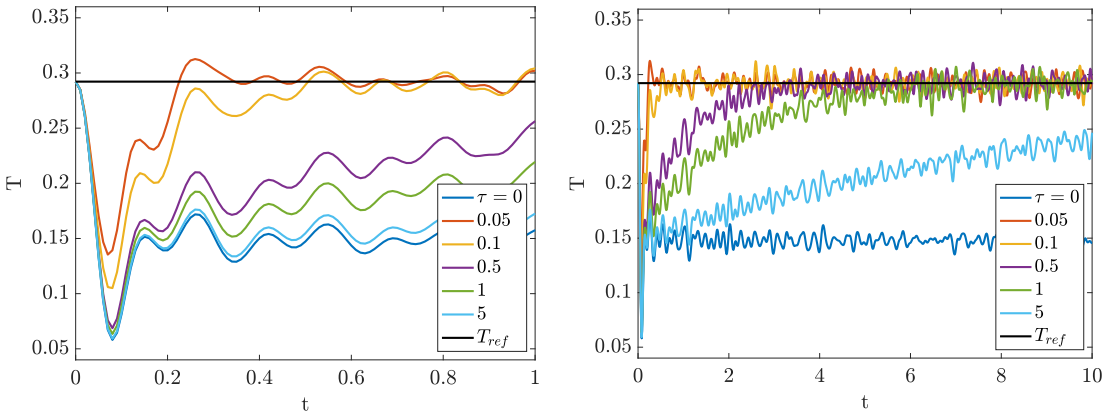

Figure 1: *Berendsen Thermostat – The rise time's dependence on the parameter* τ. After generating velocities corresponding to $T = T_{\text{ref}}$ we see an initial undershoot $\forall\tau$ (left), after which all $\tau \neq 0$ fluctuate around T_{ref} (right). A smaller non-zero τ-value gives both a smaller undershoot and shorter rise time.

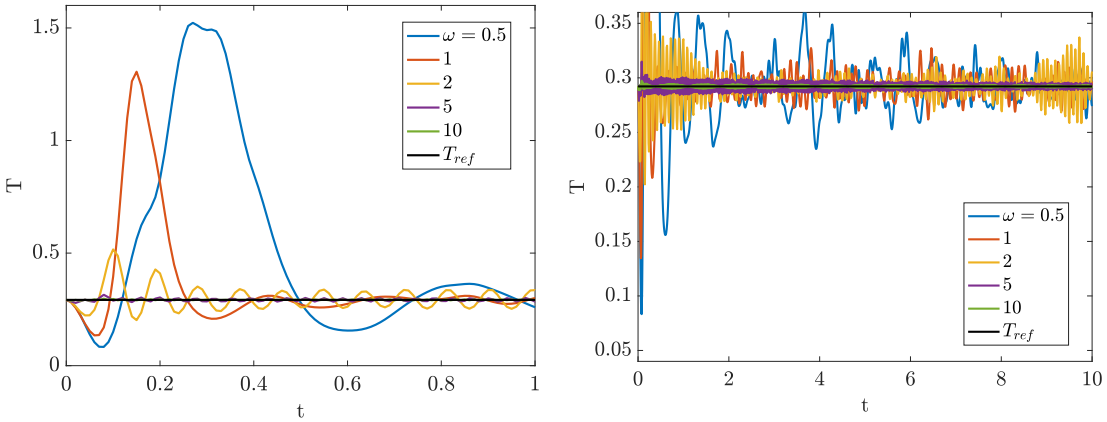

Figure 2: *Nosé-Hoover Thermostat – The rise time's dependence on the parameter* ω. After generating velocities corresponding to $T = T_{\text{ref}}$ we see an initial undershoot and a following overshoot $\forall\tau$ (left), after which all ω oscillate around T_{ref} (right). A higher ω-value gives both a smaller undershoot and overshoot and a shorter rise time.

ing the two first sets, as these correspond to times before the thermostats have begun working properly (see Fig. 1 and 2), and calculating σ_T for the other 10 sets, we obtain Fig. 3. The error bars correspond to the standard deviation of the ten calculated temperature fluctuations using the different sets.

Although one might as well have used the whole simulation (iterations 10000-60000) to calculate just one value of σ_T, it would just be the same average value as we already calculated with this method. Besides, with this method, we also captures how much the fluctuation itself fluctuates.

One can note that a smaller value of τ and a higher value of ω result in less fluctuations. Also, as the error bars get smaller with decreasing τ and increasing ω, it seems that these values are most robust.

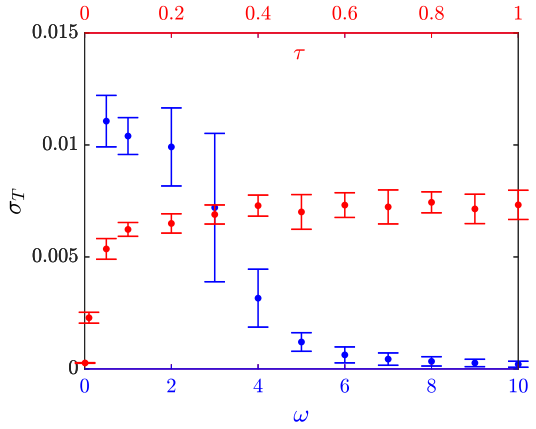

Figure 3: The temperature fluctuations σ_T in the BT (red) and the NHT (blue). To achieve low fluctuations, a low τ in the BT and a high ω in the NHT should be used.

3.2 T Step Response

To find out how well the system responds to a small sudden change in the reference temperature, and how the choice of thermostat and time parameter values τ/ω matter, a run of 20000 iterations was carried out. The reference temperature was changed from $T_{\text{ref}} = 0.29216$ (in reduced units) to $2T_{\text{ref}}$ in the middle of the simulation.

For the Berendsen thermostat, a lower value for τ results in a shorter rise time to the new reference temperature, as seen in Fig. 4. It can be seen that $\tau = 0.001$ yields a rise time of $t_{\text{rise}} \approx 0.01$, which is significantly faster than for higher τ-values.

For the Nosé-Hoover thermostat, $\omega = 2$ results in a short rise time than $\omega = 1$, which in turn is faster than $\omega = 0.5$, as shown to the left in Fig. 5. However, both $\omega = 5$ and $\omega = 10$ give unwanted oscillative behaviour, seen to the right in Fig. 5. A too high value of ω is therefore not desired, and

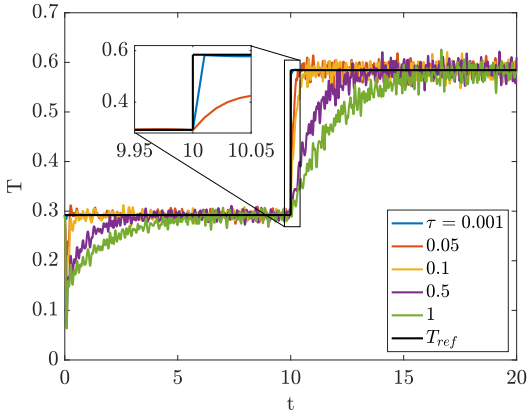

Figure 4: *Berendsen Thermostat – Step response.* A smaller value of the time parameter τ yields a much faster response, i.e. shorter rise time, to the new T_{ref}.

the reason behind this is the high absolute value of the derivative of ξ that it causes. In turn, this causes ξ, and subsequently the velocities and thus the temperature, to change so rapidly that the system goes beyond the reference temperature (in alternative directions) during each iteration.

In addition, compared to the BT, the NHT yields a significant overshoot as a consequence of the step response. In all three mentioned cases with ω that don't give oscillations, T exceeds $T = 1.0$, despite the new reference only being $T_{\text{ref}} = 0.58432$.

Another run was made to see if the size of the sudden temperature change would have any effect on the oscillations. With the step $T_{\text{ref}} \mapsto 1.2T_{\text{ref}}$ the same oscillative behaviour for $\omega = 5$ lingers. When using the NHT one, therefore, needs to choose a high ω, but not too high. In the BT-case one can choose a very small τ without any consequences in this respect.

3.3 Velocity Distribution Function

To investigate the obedience of the velocities with the Maxwell-Boltzmann distribution (MBD) the velocities after letting the thermostat work for 10000 steps are extracted. Unfortunately, this results in a small sample of only 500 velocities, so instead the velocities for the last 100 iterations are gathered.

To be able to compare the MBD with the velocities from the simulation, either the MBD needs to be converted to reduced units, or the velocities need to be converted back to unreduced units.

Using that the MBD for the speed $v = |\bar{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ has the form

$$f(v) = \sqrt{\frac{2}{\pi}} \left(\frac{m}{k_B T} \right)^{3/2} v^2 \exp\left[-\frac{mv^2}{2k_B T} \right], \quad (14)$$

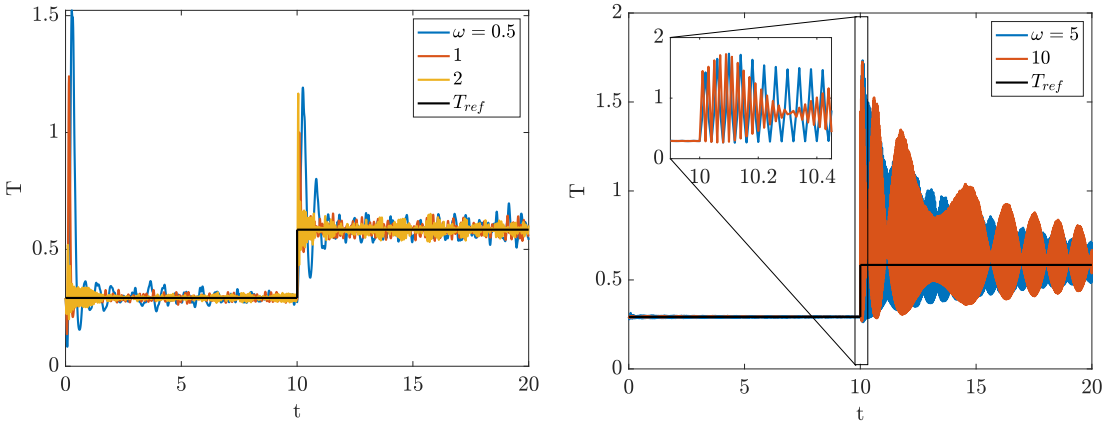

Figure 5: *Nosé-Hoover Thermostat – Step response.* A larger value of the time parameter ω yields a faster response, i.e. shorter rise time, to the new T_{ref} (left). A too high value of ω, however, results in an unwanted oscillative behaviour of the thermostat (right).

in non-reduced units, one can derive (See Appendix 5.2) the corresponding distribution function in reduced units,

$$f_{RU}(v) = \sqrt{\frac{2}{\pi}} \left(\frac{1}{T}\right)^{3/2} v^2 \exp\left[-\frac{v^2}{2T}\right], \qquad (15)$$

where T and v now are expressed in reduced units. The comparison between the distribution and the last 100 velocities is shown in Fig. 6.

It is evident that in both cases the MBD accurately describes the speed distribution. Qualitatively, it looks like there is a larger discrepancy for $\tau = 1$ for the BT, but this is not shown with any statistical significance. The velocities in the NHT definitely follows the MBD, although not as well as BT does. In the NHT, the speed distribution of particles seems to be unaffected by the values of ω.

In Appendix 5.3 one can find the corresponding distributions when only using the data from one iteration. As expected, it doesn't follow the MBD as well, which is an effect of the small sample. Moreover, in Appendix 5.4 the velocities are converted to non-reduced units and typical values of a few hundred m/s are found. Of course, the agreement between MBD and simulations is evident here as well.

3.4 Heat Capacities

Firstly, the reference temperature is changed to $T_{\text{ref}} = 0.70952$ and the argon solid state thus becomes liquid. The experimental value for the heat capacity $C_{V,\text{exp}} = 2.78\ [k_B]$ for the corresponding temperature of 85K can now be compared.

In order to extract the heat capacities, simulations of 500000 steps are performed. The long sim-

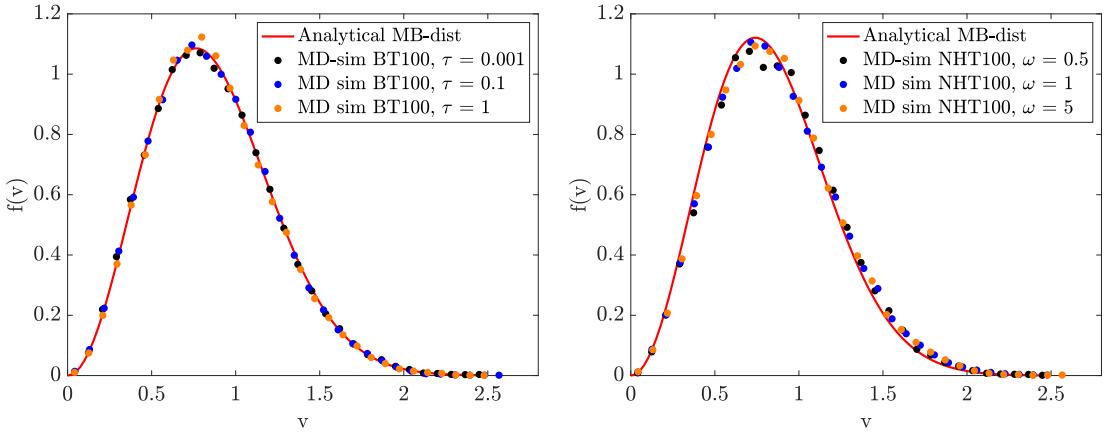

Figure 6: *Maxwell-Boltzmann Distribution for the speed of the particles.* The molecular dynamics simulation (MD-sim) with the Berendsen thermostat (BT) and the Nosé-Hoover thermostat (NHT) have been used to extract the 500 velocities from the last 100 iterations (hence BT100/NHT100). The agreement between analytical MD-distribution and simulation is shown for the BT to the left, and for the NHT to the right. In both cases the distribution seems to be followed, although slightly better in the BT-case.

ulation time allows for division into sub-intervals of 10000 steps. No generation of velocities is now made; instead, the simulations are started from the final coordinates from a previous simulation in the same state since this avoids the initial dip seen in Fig. 1 and 2.

A well-known relationship for the energy fluctuations in the canonical ensemble is

$$\langle E^2\rangle - \langle E\rangle^2 = k_B T^2 \frac{\partial \langle E\rangle}{\partial T} = k_B T^2 C_V, \quad (16)$$

from which we can find the heat capacity C_V. The corresponding expression in non-reduced units is

$$C_V = \frac{\langle E^2\rangle - \langle E\rangle^2}{NT^2} \quad [k_B], \quad (17)$$

where we scale the energy-values from the simulation with $1/N$ to obtain the heat capacity per particle. Extracting the data and calculating the energy fluctuations from the different parts of the simulation, thus, results in a sample of C_V:s. The result is presented in histogram-form in Fig. 8.

Due to the disagreement between the simulated heat capacity and the experimental value, the BT with $\tau = 1$ is only used during one simulation. NHT with $\omega = 1$, however, is seen to overlap much better with the experimental value and three long runs are performed. The average value of $C_V = 2.6639$ $[k_B]$ resembles the experimental value of $C_{V,\text{exp}} = 2.78$ $[k_B]$. Taking the standard deviation of 0.3687 into account, the MD-simulation with the NHT captures the experimental value.

Another way to calculate the heat capacity is to perform the simulation in the microcanonical NVE-ensemble, and use the temperature fluctuations as in

$$C_V = \frac{3}{2}\frac{1}{1 - \frac{3}{2}N\frac{\langle (T-\langle T\rangle)^2\rangle}{\langle T\rangle^2}} \quad [k_B]. \quad (18)$$

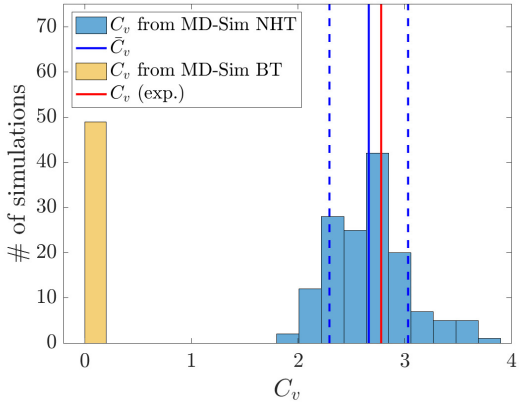

Figure 8: *Heat capacity calculated from the canonical ensemble's expression.* The agreement between the calculated C_V (using Eq. (16)) from molecular dynamics simulations (MD-sim) with the Berendsen thermostat (BT) and the Nosé-Hoover thermostat (NHT) respectively is shown. The C_V from the BT does not agree with the experimental value and less simulations are performed. In the NHT-case the agreement is evident; the average value $\bar{C}_V = 2.6639$ $[k_B]$ agrees with the experimental value $C_{V,\text{exp}} = 2.78$ $[k_B]$. The dashed blue lines represent the standard deviation for the NHT-data from its average value.

Despite the simulations being performed in an NVT-ensemble we can try to apply this equation to calculate the heat capacity. The result is presented in Fig. 7 for different values of the time parameters. For the previously used values of $\tau \in [0.01, 1]$ in the BT the agreement is quite good, while the agreement in the NHT-case with $\omega \in [0.1, 10]$ substantially disagree with the experimental $C_{V,\text{exp}}$. In both cases, we see that when $\tau \to \infty$ and $\omega \to 0$, respectively, corresponding

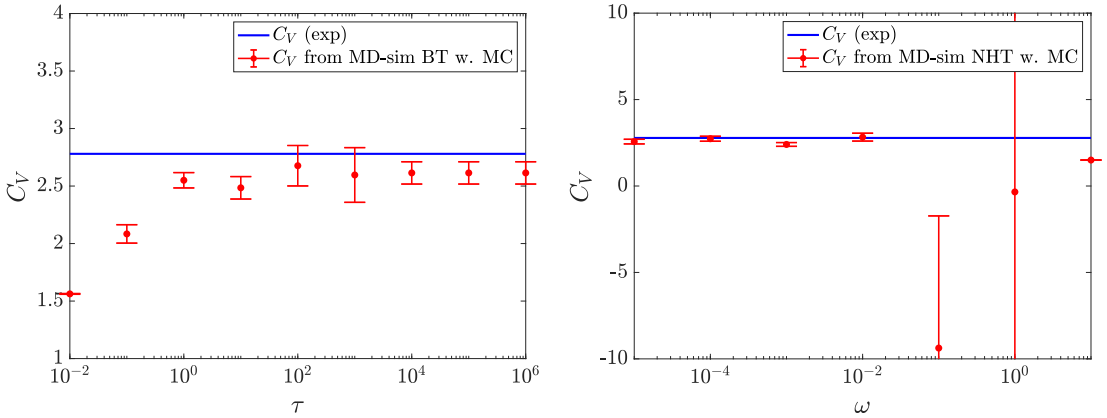

Figure 7: *Heat capacity calculated from the microcanonical ensemble's expression.* In the Berendsen thermostat (BT, left) all values of τ gives a C_V that agrees fairly with the experimental value. In the Nosé-Hoover thermostat (NHT, right), however, we need very small values of ω to get close to $C_{V,\text{exp}}$; the previously used values of $\omega \in [0.1, 10]$ give large discrepancies.

to making the thermostats inactive, the simulated system goes towards the microcanonical ensemble, with the C_V-values converging towards a value near $C_{V,\text{exp}}$. In both these limits, the temperature fluctuations will adjust until they have reached the appropriate value of a microcanonical ensemble.

4 Discussion & Conclusion

It has been shown that by using thermostats, in particular the Berendsen thermostat (BT) and the Nosé-Hoover thermostat (NHT), one can achieve a constant temperature, albeit with fluctuations around this value. The two thermostats' performance has been investigated and in both cases, it is clearly dependent on the time parameters τ (in BT) and ω (in NHT). A low value of τ and a high value of ω, respectively, results in a thermostat that quickly gets the system to an equilibrated state around T_{ref} after generating velocities at this temperature. The same goes for the temperature fluctuations, which are comparable in size in both thermostats, and that goes down with a low value of τ and a high value of ω.

In terms of how well the system responds to a step in T_{ref}, one ought to have a low value of τ when using the BT. Unarguable, low values of τ seems to be the best choice. For NHT, however, if one chooses a too high value of ω then the system oscillates extensively and the system doesn't thermalize. Therefore, it's a balancing act to find the best ω; for small thermal fluctuations one wants a high value, but not too high as this leads to oscillations.

Nonetheless, both systems achieve a Maxwellian speed distribution, as expected. In this respect, both thermostats work properly and there is no major difference.

Regarding the heat capacity, there is a substantial difference between the BT and the NT. The BT does not allow for the energy to fluctuate as it should in a canonical ensemble. On the contrary, the NHT manages to rigorously generate the canonical ensemble thermodynamics, with the correct energy fluctuations.

On the other hand, it is shown that the BT can be used to effectively drive the NVT-system close to a microcanonical ensemble, as the thermal fluctuation equation gives a reasonable value for the heat capacity regardless of the time parameter value. Both thermostats can, however, be pushed towards a microcanonical ensemble by letting $\tau \to \infty$ or $\omega \to 0$, respectively, as this inactivates the thermostats.

To conclude, the two thermostats are performing well in different situations. To represent a realistic canonical ensemble, one needs to use the NHT, but for a quick equilibration to a new reference temperature, one should use the BT. In other words, one should use different thermostats for different purposes. Optimally, one should use both in a molecular dynamics simulation and switch between them; one can use the NHT as long as the system starts near equilibrium and whenever a step in T_{ref} occurs, one ought to shift to and use the BT.

References

[1] Daan Frenkel and Berend Smit. Molecular Dynamics in Various Ensembles. *Understanding Molecular Simulation*, pages 139–163, 2002.

[2] Philippe H. Hünenberger. Thermostat algorithms for molecular dynamics simulations. *Advances in Polymer Science*, 173:105–147, 2005.

[3] M.S Shell. Advanced Molecular Dynamics. *Advanced molecular dynamics techniques. Lecture Notes from Course: Principles of modern molecular simulation methods at University of California Santa Barbara, Department of Chemical Engineering* ., pages 1–11, 2009.

[4] Johannes Niskanen and Henning Henschel. Controlling temperature. *Lecture Notes from Course: Molecular Dynamics Simulations at University of Helsinki, Physics Department.*, 3(29):13, 2001.

[5] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984.

5 Appendix

5.1 Reduced units

Quantity	Reduced unit	Relation to SI
Length	r^*	$r\sigma^{-1}$
Energy	E^*	$E\epsilon^{-1}$
Temperature	T^*	$k_BT\epsilon^{-1}$
Time	t^*	$t\sigma^{-1}\sqrt{\epsilon/M}$
Mass	m^*	mM^{-1}
Force	F^*	$F\sigma\epsilon^{-1}$
Density	ρ^*	$N\sigma^3V^{-1}$
Pressure	p^*	$p\sigma^3\epsilon^{-1}$
Velocity	v^*	$v\sqrt{M/\epsilon}$

Table 1: Reduced units.

The used parameters are $M = 39.948u$ (where $u = 1.6605 \cdot 10^{-27}$ kg), $\sigma = 3.40$, and $\epsilon = 1.654 \cdot 10^{-21}$ J.

5.2 Velocity Distribution Function

We start from

$$f(v) = \sqrt{\frac{2}{\pi}} \left(\frac{m}{k_BT}\right)^{3/2} v^2 \exp\left[-\frac{mv^2}{2k_BT}\right], \tag{19}$$

in non-reduced units. From Appendix 5.1 we have that

$$\begin{aligned} k_BT &= T^*\epsilon \\ m &= m^*M \\ v &= v^*\sqrt{\epsilon/M}. \end{aligned} \tag{20}$$

We use these to simplify Eq. (19) as

$$\begin{aligned} &\sqrt{\frac{\pi}{2}} f(v) = \left(\frac{m}{k_BT}\right)^{3/2} v^2 \exp\left[-\frac{mv^2}{2k_BT}\right] = \left(\frac{m^*M}{T^*\epsilon}\right)^{3/2} v^{*2} \frac{\epsilon}{M} \exp\left[-\frac{m^*Mv^{*2}\epsilon}{2T^*\epsilon M}\right] = \\ &= \left(\frac{1}{T^*}\right)^{3/2} v^{*2} \left(\frac{M}{\epsilon}\right)^{1/2} \exp\left[-\frac{v^{*2}}{2T^*}\right], \end{aligned} \tag{21}$$

where we also used that $m^* = 1$ since the mass m for each argon particle is exactly the mass M.

In addition, we need to make the substitution $dv = dv^*\sqrt{\epsilon/M}$ to modify $f(v)dv$ from being the probability to find a particle with a velocity in the interval $[v, v+dv]$ to find one in the interval $[v^*, v^*+dv^*]$. By doing this substitution we maintain the vital probability relationship

$$\int_0^\infty f(v)dv = 1 \implies \int_0^\infty f(v^*)dv^* = 1. \tag{22}$$

If we use the expressions for $\sqrt{\frac{\pi}{2}} f(v)$ and dv respectively, and drop the superscript $*$ for convenience, the distribution function in reduced units becomes

$$\boxed{f_{RU}(v) = \sqrt{\frac{2}{\pi}} \left(\frac{1}{T}\right)^{3/2} v^2 \exp\left[-\frac{v^2}{2T}\right]} \quad , \tag{23}$$

where T and v now are expressed in reduced units.

5.3 MB with only 1 iteration

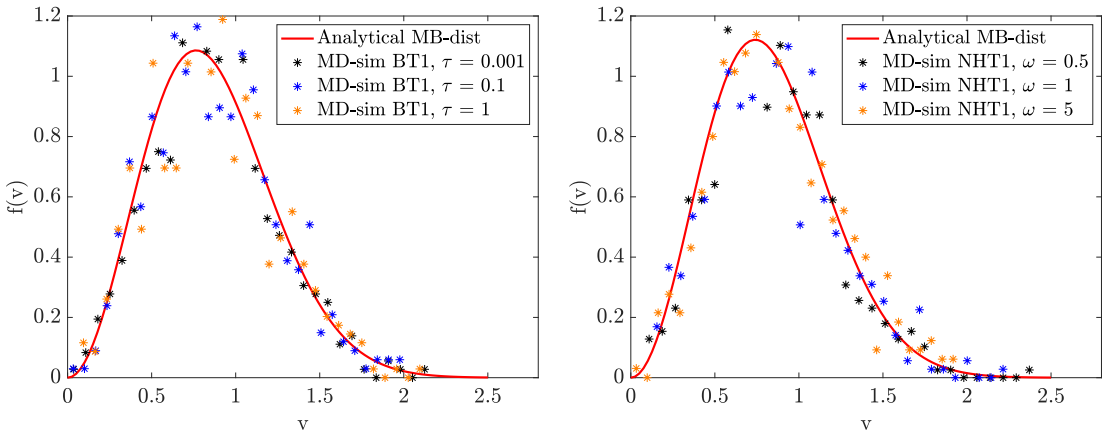

Figure 9: *Maxwell-Boltzmann Distribution for the speed of the particles.* The molecular dynamics simulation (MD-sim) with the Berendsen thermostat (BT) and the Nosé-Hoover thermostat (NHT) have been used to extract the 500 velocities from the *single* last iterations (hence BT1/NHT1). The agreement between analytical MD-distribution and simulation is shown for the BT to the left, and for the NHT to the right. In both cases the distribution seems to be followed to some extent, although there might be a problem with a small sample of only 500 speeds.

5.4 MB in non-reduced units

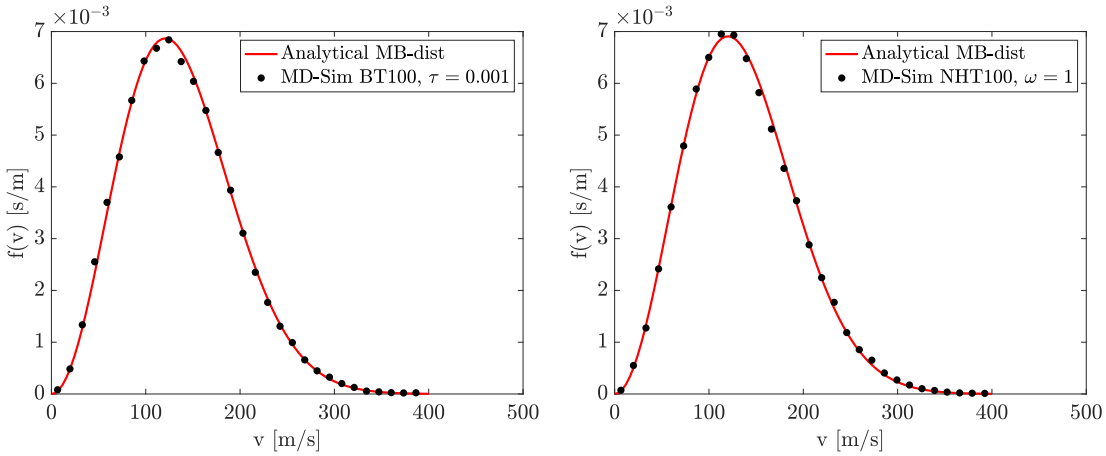

Figure 10: *Maxwell-Boltzmann Distribution for the speed of the particles.* The differences from Fig. 6 are that everything is plotted in non-reduced units and that only one value of the time parameters was used.

Code

Analyzing data – MATLAB

Velocities

Here is one example of the code that's been used to do the data analysis. This script was used to generate the Maxwellian-distribution-figures: Fig. 6, 9, & 10.

```
% FOR NH-thermostat

set(groot,'defaultAxesTickLabelInterpreter','latex');
% set(groot,'defaultLegendInterpreter','latex');

E_v = {'energy_w05.ene';'energy_w1.ene';'energy_w5.ene'};
v2_v = {'v2_w05.dat';'v2_w1.dat';'v2_w5.dat'};
colors = {'k','b',[1,0.5,0]};

for i = 1:size(E_v,1)
    E = load(E_v{i});
    v2 = load(v2_v{i});
    c = colors{i}

    v2_lastiteration = v2(49501:50000);
    eps = 1.654*10^(-21);
    u = 1.6605402*10^(-27);
    M = 39.948*u; %m_argon
    T = E(end,10);
    kB = 1.380649*10^(-23);

    figure(1)
    % Create speed distribution function: speed = sqrt(v2)
    N = 30;
    hist = histogram(sqrt(v2),N,'Normalization','probability');
    hold on
    x_hist = hist.BinEdges(1,1:N)+hist.BinWidth/2;
    y_hist = hist.Values./hist.BinWidth;
    hist = histogram(sqrt(v2_lastiteration),N,'Normalization','probability');
    x_hist_last = hist.BinEdges(1,1:N)+hist.BinWidth/2;
    y_hist_last = hist.Values./hist.BinWidth;
    hold off

    figure(2)
    % Maxwell-Boltzmandistribution in reduced units
    % https://en.wikipedia.org/wiki/Maxwell%E2%80%93Boltzmann_distribution
    if i == 1
        f = @(v) sqrt(2/pi)*(1/T)^(3/2)*v.^2.*exp(-v.^2/(2*T));
        v = linspace(0,2.5,100);
        plot(v,f(v),'r','linewidth',2);
        hold on
    end
    plot(x_hist,y_hist,'.','color',c,'markersize',20)


    figure(3)
    if i == 1
        plot(v,f(v),'r','linewidth',2);
        hold on
    end
    plot(x_hist_last,y_hist_last,'*','color',c,'markersize',8)
end

figure(1)
legend('Last 100 iterations','Last iteration')
set(gca,'fontsize',16);
xlabel('v','interpreter','latex','fontsize',20);
ylabel('f(v)','interpreter','latex','fontsize',20);

figure(2)
axis([0 2.8 0 1.2]);
legend('Analytical MB-dist',...
    'MD-sim NHT100, $\omega$ = 0.5',...
```

```
    'MD sim NHT100, $\omega$ = 1',...
    'MD sim NHT100, $\omega$ = 5','interpreter','latex')
set(gca,'fontsize',20);
xlabel('v','interpreter','latex','fontsize',20);
ylabel('f(v)','interpreter','latex','fontsize',20);

print -depsc2 '../../PicReport/thermostat_NH_100iter.eps'

figure(3)
axis([0 2.8 0 1.2]);
legend('Analytical MB-dist',...
    'MD-sim NHT1, $\omega$ = 0.5',...
    'MD-sim NHT1, $\omega$ = 1',...
    'MD-sim NHT1, $\omega$ = 5',...
    'interpreter','latex')

set(gca,'fontsize',20);
xlabel('v','interpreter','latex','fontsize',20);
ylabel('f(v)','interpreter','latex','fontsize',20);

print -depsc2 '../../PicReport/thermostat_NH_1iter.eps'

%% In non-reduced units
v2 = load('v2_w1.dat');
E = load('energy_w1.ene');
v2_lastiteration = v2(49501:50000);

figure(3)

hist = histogram(sqrt(v2)*sqrt(eps/M),N,'Normalization','probability');
x_hist = hist.BinEdges(1,1:N)+hist.BinWidth/2;
y_hist = hist.Values./hist.BinWidth;

T_nonred = T/kB*eps;
f = @(v) (M/(2*pi*kB*T_nonred))^(3/2)*4*pi*v.^2.*exp(-M*v.^2/(2*kB*T_nonred));
v = linspace(0,400,100);
plot(v,f(v),'r','linewidth',2);
hold on
plot(x_hist,y_hist,'.k','markersize',20)

axis([0 500 0 0.007]);
legend('Analytical MB-dist','MD-Sim NHT100, $\omega = 1$','interpreter','latex')
set(gca,'fontsize',20);
xlabel('v [m/s]','interpreter','latex','fontsize',20);
ylabel('f(v) [s/m]','interpreter','latex','fontsize',20);

print -depsc2 '../../PicReport/thermostat_NH_100iter_nonreduced.eps'
```

Time parameters

This script was used to generate Fig. 3.

```
%% Fluctuations for BT and NHT in same plot

Tref = 0.29216;
fig = figure(2)
set(groot,'defaultAxesTickLabelInterpreter','latex');

E_v = {'energy_w05_long.ene';'energy_w1_long.ene';'energy_w2_long.ene';...
    'energy_w3_long.ene';'energy_w4_long.ene';'energy_w5_long.ene';...
    'energy_w6_long.ene';'energy_w7_long.ene';'energy_w8_long.ene';...
    'energy_w9_long.ene';'energy_w10_long.ene'};

w_v = [0.5,1,2,3,4,5,6,7,8,9,10];
for i = 1:size(E_v,1)
    %Plot_color=cmap(i*30,:);
    E = load(E_v{i});
    step = E(:,1);
    t = E(:,2);
    T = E(:,10);
    Nsteps = length(T);
```

```
    N = 500;
    kB = 1.380649*10^(-23);

    % Time average
    avT = sum(T)/Nsteps;

    % RMSD (Fluctuations)
    msdT = (sum((T-avT).^2)/Nsteps);

    % RMSD (Fluctuations)
    Cv = (3/2)*1/(1-(3/2)*N*msdT/(avT^2));

    average_box = mean(L);

    % Plot fluctuations
    Fluct = zeros(10,1);
    for j = 1:10
       % Look at the t-intervals 10001-15000,15001-20000 etc
       start_ind = 1001+(j-1)*500;
       end_ind = 1000+j*500;

       t = E(start_ind:end_ind,2);
       T = E(start_ind:end_ind,10);

       % Calculate Temp-fluctuations around Tref
       sigT = sqrt(1/(size(t,1)-1) * sum((T-Tref).^2));
       Fluct(j) = sigT;
    end

    % Calculate average & fluct for all Fluctuations of 5000-steps
    avFluct = mean(Fluct);
    sigCv = sqrt(1/10*sum((Fluct-avFluct).^2));

    % Plot
    e = errorbar(w_v(i),avFluct,sigCv);
    e.Marker = '.';
    e.MarkerSize = 20;
    e.Color = 'b';
    e.CapSize = 20;
    e.LineWidth = 1.5;
    hold on

end

ax1 = gca; % current axes
axis([0 10 0 0.015])
ax1.XColor = 'b';
ax1.YColor = 'k';
set(gca,'fontsize',20);
xlabel('$\omega$','interpreter','latex','fontsize',24);
ylabel('$\sigma_T$','interpreter','latex','fontsize',24);

ax1_pos = ax1.Position; % position of first axes
ax2 = axes('Position',ax1_pos);

addpath '/Users/ludde/Documents/KTH/ r5 /CompPhys/Final ...
    Project/CODE_work/BERENDSEN_solid/Energy_files_tau'
E_v = {'energy_tau1_long.ene';'energy_tau09_long.ene';'energy_tau08_long.ene';...
    'energy_tau07_long.ene';'energy_tau06_long.ene';'energy_tau05_long.ene';...
    'energy_tau04_long.ene';'energy_tau03_long.ene';'energy_tau02_long.ene';...
    'energy_tau01_long.ene';'energy_tau005_long.ene';'energy_tau001_long.ene';...
    'energy_tau0001_long.ene'};

tau_v = [1;0.9;0.8;0.7;0.6;0.5;0.4;0.3;0.2;0.1;0.05;0.01;0.001];
for i = 1:size(E_v,1)
    %Plot_color=cmap(i*30,:);
    E = load(E_v{i});
    step = E(:,1);
    t = E(:,2);
    Epot = E(:,3);
    Ekin = E(:,4);
    Vir = E(:,5);
    L = E(:,6);
```

```
92      Pkin = E(:,7);
93      Pvir = E(:,8);
94      Ptot = E(:,9);
95      T = E(:,10);
96      Nsteps = length(T);
97      N = 500;
98      kB = 1.380649*10^(-23);
99
100     % Time average
101     avT = sum(T)/Nsteps;
102
103     % RMSD (Fluctuations)
104     msdT = (sum((T-avT).^2)/Nsteps);
105
106     % RMSD (Fluctuations)
107     Cv = (3/2)*1/(1-(3/2)*N*msdT/(avT^2));
108
109
110     % Plot fluctuations
111     Fluct = zeros(10,1);
112     for j = 1:10
113        % Look at the t-intervals 10001-15000,15001-20000 etc
114        start_ind = 1001+(j-1)*500;
115        end_ind = 1000+j*500;
116
117        t = E(start_ind:end_ind,2);
118        T = E(start_ind:end_ind,10);
119
120        % Calculate Temp-fluctuations around Tref
121        sigT = sqrt(1/(size(t,1)-1) * sum((T-Tref).^2));
122        Fluct(j) = sigT;
123     end
124
125     % Calculate average & fluct for all Fluctuations of 5000-steps
126     avFluct = mean(Fluct);
127     sigCv = sqrt(1/10*sum((Fluct-avFluct).^2));
128
129     % Plot
130     e = errorbar(tau_v(i),avFluct,sigCv,'Parent',ax2,'Color','k');
131     e.Marker = '.';
132     e.MarkerSize = 20;
133     e.Color = 'r';
134     e.CapSize = 20;
135     e.LineWidth = 1.5;
136     hold on
137
138 end
139
140 set(gca,'XAxisLocation','top',...
141     'YAxisLocation','left',...
142     'Color','none')
143 axis([0 1 0 0.015])
144 set(gca,'fontsize',20);
145 xlabel('$\tau$','interpreter','latex','fontsize',24);
146
147 % Create textbox
148 annotation(fig,'textbox',...
149     [0.502785714285714 0.85 0.054357142857143 0.0714285714285735],...
150     'String','\tau',...
151     'Color','r',...
152     'Fontsize',22,...
153     'Edgecolor','None',...
154     'FitBoxToText','off');
155
156 set(gca,'Xcolor',[1 0 0]); % Set RGB value to what you want
157 plot([0,10],[0,0],'b','linewidth',1.2)
158 plot([0,10],[0.015,0.015],'r','linewidth',1.2)
159 plot([0.2,0.2],[0,0.0002],'b')
160 plot([0.4,0.4],[0,0.0002],'b')
161 plot([0.6,0.6],[0,0.0002],'b')
162 plot([0.8,0.8],[0,0.0002],'b')
163
164
```

```
% print -depsc2 '../../PicReport/thermostat_NHBT_wTAU_analysis.eps'
```

Molecular Dynamics – FORTRAN

Berendsen Thermostat

```
*      LAMBDA - scale factor for Berendsen temp. coupling
       IF(TEMP.GT.0.0 .AND. TAUT.GT.0.0) THEN
*          LAMBDA = (1.0 + DT/TAUT*(TREF/TEMP-1.0))**(0.5)
           XI = -1.0/(2.0*TAUT)*(TREF/TEMP-1.0)
       ELSE
*          LAMBDA = 1.0
           XI = 0
       ENDIF
```

Nosé-Hoover Thermostat

```
*      LAMBDA - scale factor for Berendsen temp. coupling
       IF(TEMP.GT.0.0 .AND. TAUT.GT.0.0) THEN
           XI = XI + (-NATOMS*TAUT*TAUT*(TREF/TEMP-1.0))*DT
       ELSE
           XI = 0
       ENDIF
```

Updating velocities

```
       DO 15, I = 1,NATOMS
*      Update and scale velocities
*           VX(I) = LAMBDA*(VX(I) + FX(I)*DT)
*           VY(I) = LAMBDA*(VY(I) + FY(I)*DT)
*           VZ(I) = LAMBDA*(VZ(I) + FZ(I)*DT)
*      or equivalently:
            VX(I) = VX(I) + (FX(I) - XI*VX(I))*DT
            VY(I) = VY(I) + (FY(I) - XI*VY(I))*DT
            VZ(I) = VZ(I) + (FZ(I) - XI*VZ(I))*DT
```

Making a step in T_{ref}

```
           IF(TIME/(NSTEPS*DT).EQ.0.5) THEN
             TREF = 1.2*TREF
*             PRINT *,"TREF changed to 2*TREF"
           END IF
```

Extracting data, e.g., velocities

```
*      Let elements 1-500 be for one time step, 500-1000 the next ...
     time step etc
       NUM_ITER = 100
       DO 11, I = 1,500*NUM_ITER
          V2TOT(I) = .0
  11   CONTINUE
```

```
*      Calculate Velocity squared for the last NUM_ITER iterations
*      V2TOT[1-500] = the squared vel forall part. at NUM_ITER:th ... last iteration
*      V2TOT[501-1000] = -||- (NUM_ITER-1):th to last iteration
*      V2TOT[1+500*(NUM_ITER-1)-500+500*(NUM_ITER-1)] = -||- during ... the last iter

       DO 41 , I = 1,NUM_ITER
         NF = NFRAMES - (NUM_ITER-I)
         DO 40, K = 1,NATOMS
           V2 = VX(K,NF)*VX(K,NF)+VY(K,NF)*VY(K,NF)+VZ(K,NF)*VZ(K,NF)
           V2TOT(K+500*(I-1)) = V2
40       CONTINUE
41     CONTINUE
```

```
*      Write v2
       OPEN(UNIT=92,FILE='v2.dat')
       DO 50, I = 1,NATOMS*NUM_ITER
          WRITE(92,'(1F8.3)')V2TOT(I)
 50    CONTINUE
       CLOSE(92)
```