3.4.8 Temperature coupling

While direct use of molecular dynamics gives rise to the NVE (constant number, constant volume, constant energy ensemble), most quantities that we wish to calculate are actually from a constant temperature (NVT) ensemble, also called the canonical ensemble. GROMACS can use the *weak-coupling* scheme of Berendsen [25], stochastic randomization through the Andersen thermostat [26], the extended ensemble Nosé-Hoover scheme [27, 28], or a velocity-rescaling scheme [29] to simulate constant temperature, with advantages of each of the schemes laid out below.

There are several other reasons why it might be necessary to control the temperature of the system (drift during equilibration, drift as a result of force truncation and integration errors, heating due to external or frictional forces), but this is not entirely correct to do from a thermodynamic standpoint, and in some cases only masks the symptoms (increase in temperature of the system) rather than the underlying problem (deviations from correct physics in the dynamics). For larger systems, errors in ensemble averages and structural properties incurred by using temperature control to remove slow drifts in temperature appear to be negligible, but no completely comprehensive comparisons have been carried out, and some caution must be taking in interpreting the results.

Berendsen temperature coupling

The Berendsen algorithm mimics weak coupling with first-order kinetics to an external heat bath with given temperature T_0. See ref. [30] for a comparison with the Nosé-Hoover scheme. The effect of this algorithm is that a deviation of the system temperature from T_0 is slowly corrected according to:

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{T_0 - T}{\tau} \tag{3.44}$$

which means that a temperature deviation decays exponentially with a time constant τ. This method of coupling has the advantage that the strength of the coupling can be varied and adapted to the user requirement: for equilibration purposes the coupling time can be taken quite short (*e.g.* 0.01 ps), but for reliable equilibrium runs it can be taken much longer (*e.g.* 0.5 ps) in which case it hardly influences the conservative dynamics.

The Berendsen thermostat suppresses the fluctuations of the kinetic energy. This means that one does not generate a proper canonical ensemble, so rigorously, the sampling will be incorrect. This error scales with $1/N$, so for very large systems most ensemble averages will not be affected significantly, except for the distribution of the kinetic energy itself. However, fluctuation properties, such as the heat capacity, will be affected. A similar thermostat which does produce a correct ensemble is the velocity rescaling thermostat [29] described below.

The heat flow into or out of the system is affected by scaling the velocities of each particle every step, or every n_{TC} steps, with a time-dependent factor λ, given by:

$$\lambda = \left[1 + \frac{n_{\mathrm{TC}}\Delta t}{\tau_T}\left\{\frac{T_0}{T(t - \frac{1}{2}\Delta t)} - 1\right\}\right]^{1/2} \tag{3.45}$$

The parameter τ_T is close, but not exactly equal, to the time constant τ of the temperature coupling (eqn. 3.44):

$$\tau = 2 C_V \tau_T / N_{df} k \tag{3.46}$$

where C_V is the total heat capacity of the system, k is Boltzmann's constant, and N_{df} is the total number of degrees of freedom. The reason that $\tau \neq \tau_T$ is that the kinetic energy change caused by scaling the velocities is partly redistributed between kinetic and potential energy and hence the change in temperature is less than the scaling energy. In practice, the ratio τ/τ_T ranges from 1 (gas) to 2 (harmonic solid) to 3 (water). When we use the term "temperature coupling time constant," we mean the parameter τ_T. **Note** that in practice the scaling factor λ is limited to the range of 0.8 $<=\lambda<=$ 1.25, to avoid scaling by very large numbers which may crash the simulation. In normal use, λ will always be much closer to 1.0.

Velocity-rescaling temperature coupling

The velocity-rescaling thermostat [29] is essentially a Berendsen thermostat (see above) with an additional stochastic term that ensures a correct kinetic energy distribution by modifying it according to

$$\mathrm{d}K = (K_0 - K)\frac{\mathrm{d}t}{\tau_T} + 2\sqrt{\frac{KK_0}{N_f}}\frac{\mathrm{d}W}{\sqrt{\tau_T}}, \tag{3.47}$$

where K is the kinetic energy, N_f the number of degrees of freedom and $\mathrm{d}W$ a Wiener process. There are no additional parameters, except for a random seed. This thermostat produces a correct canonical ensemble and still has the advantage of the Berendsen thermostat: first order decay of temperature deviations and no oscillations. When an NVT ensemble is used, the conserved energy quantity is written to the energy and log file.

Andersen thermostat

One simple way to maintain a thermostatted ensemble is to take an NVE integrator and periodically re-select the velocities of the particles from a Maxwell-Boltzmann distribution. [26] This can either be done by randomizing all the velocities simultaneously (massive collision) every $\tau_T/\Delta t$ steps (`andersen-massive`), or by randomizing every particle with some small probability every timestep (`andersen`), equal to $\Delta t/\tau$, where in both cases Δt is the timestep and τ_T is a characteristic coupling time scale. Because of the way constraints operate, all particles in the same constraint group must be randomized simultaneously. Because of parallelization issues, the `andersen` version cannot currently (5.0) be used in systems with constraints. `andersen-massive` can be used regardless of constraints. This thermostat is also currently only possible with velocity Verlet algorithms, because it operates directly on the velocities at each timestep.

This algorithm completely avoids some of the ergodicity issues of other thermostatting algorithms, as energy cannot flow back and forth between energetically decoupled components of the system as in velocity scaling motions. However, it can slow down the kinetics of system by randomizing correlated motions of the system, including slowing sampling when τ_T is at moderate levels (less than 10 ps). This algorithm should therefore generally not be used when examining kinetics or transport properties of the system. [31]

Nosé-Hoover temperature coupling

The Berendsen weak-coupling algorithm is extremely efficient for relaxing a system to the target temperature, but once the system has reached equilibrium it might be more important to probe a correct canonical ensemble. This is unfortunately not the case for the weak-coupling scheme.

To enable canonical ensemble simulations, GROMACS also supports the extended-ensemble approach first proposed by Nosé [27] and later modified by Hoover [28]. The system Hamiltonian is extended by introducing a thermal reservoir and a friction term in the equations of motion. The friction force is proportional to the product of each particle's velocity and a friction parameter, ξ. This friction parameter (or "heat bath" variable) is a fully dynamic quantity with its own momentum (p_ξ) and equation of motion; the time derivative is calculated from the difference between the current kinetic energy and the reference temperature.

In this formulation, the particles' equations of motion in Fig. 3.3 are replaced by:

$$\frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} = \frac{\boldsymbol{F}_i}{m_i} - \frac{p_\xi}{Q}\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t}, \tag{3.48}$$

where the equation of motion for the heat bath parameter ξ is:

$$\frac{\mathrm{d}p_\xi}{\mathrm{d}t} = (T - T_0)\,. \tag{3.49}$$

The reference temperature is denoted T_0, while T is the current instantaneous temperature of the system. The strength of the coupling is determined by the constant Q (usually called the "mass parameter" of the reservoir) in combination with the reference temperature. [1]

The conserved quantity for the Nosé-Hoover equations of motion is not the total energy, but rather

$$H = \sum_{i=1}^{N} \frac{\boldsymbol{p}_i}{2m_i} + U\left(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N\right) + \frac{p_\xi^2}{2Q} + N_f kT\xi, \tag{3.50}$$

where N_f is the total number of degrees of freedom.

In our opinion, the mass parameter is a somewhat awkward way of describing coupling strength, especially due to its dependence on reference temperature (and some implementations even include the number of degrees of freedom in your system when defining Q). To maintain the coupling strength, one would have to change Q in proportion to the change in reference temperature. For this reason, we prefer to let the GROMACS user work instead with the period τ_T of the oscillations of kinetic energy between the system and the reservoir instead. It is directly related to Q and T_0 via:

$$Q = \frac{\tau_T^2 T_0}{4\pi^2}. \tag{3.51}$$

This provides a much more intuitive way of selecting the Nosé-Hoover coupling strength (similar to the weak-coupling relaxation), and in addition τ_T is independent of system size and reference temperature.

It is however important to keep the difference between the weak-coupling scheme and the Nosé-Hoover algorithm in mind: Using weak coupling you get a strongly damped *exponential relaxation*, while the Nosé-Hoover approach produces an *oscillatory relaxation*. The actual time it takes to relax with Nosé-Hoover coupling is several times larger than the period of the oscillations that you select. These oscillations (in contrast to exponential relaxation) also means that the time constant normally should be 4–5 times larger than the relaxation time used with weak coupling, but your mileage may vary.

[1]Note that some derivations, an alternative notation $\xi_{\mathrm{alt}} = v_\xi = p_\xi/Q$ is used.

Nosé-Hoover dynamics in simple systems such as collections of harmonic oscillators, can be *non-ergodic*, meaning that only a subsection of phase space is ever sampled, even if the simulations were to run for infinitely long. For this reason, the Nosé-Hoover chain approach was developed, where each of the Nosé-Hoover thermostats has its own Nosé-Hoover thermostat controlling its temperature. In the limit of an infinite chain of thermostats, the dynamics are guaranteed to be ergodic. Using just a few chains can greatly improve the ergodicity, but recent research has shown that the system will still be nonergodic, and it is still not entirely clear what the practical effect of this [32]. Currently, the default number of chains is 10, but this can be controlled by the user. In the case of chains, the equations are modified in the following way to include a chain of thermostatting particles [33]:

$$\begin{aligned}
\frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} &= \frac{\boldsymbol{F}_i}{m_i} - \frac{p_{\xi_1}}{Q_1}\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} \\
\frac{\mathrm{d}p_{\xi_1}}{\mathrm{d}t} &= (T - T_0) - p_{\xi_1}\frac{p_{\xi_2}}{Q_2} \\
\frac{\mathrm{d}p_{\xi_{i=2\ldots N}}}{\mathrm{d}t} &= \left(\frac{p^2_{\xi_{i-1}}}{Q_{i-1}} - kT\right) - p_{\xi_i}\frac{p_{\xi_{i+1}}}{Q_{i+1}} \\
\frac{\mathrm{d}p_{\xi_N}}{\mathrm{d}t} &= \left(\frac{p^2_{\xi_{N-1}}}{Q_{N-1}} - kT\right)
\end{aligned} \tag{3.52}$$

The conserved quantity for Nosé-Hoover chains is

$$H = \sum_{i=1}^{N} \frac{\boldsymbol{p}_i}{2m_i} + U(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) + \sum_{k=1}^{M} \frac{p^2_{\xi_k}}{2Q'_k} + N_f kT\xi_1 + kT\sum_{k=2}^{M}\xi_k \tag{3.53}$$

The values and velocities of the Nosé-Hoover thermostat variables are generally not included in the output, as they take up a fair amount of space and are generally not important for analysis of simulations, but this can be overridden by defining the environment variable `GMX_NOSEHOOVER_CHAINS`, which will print the values of all the positions and velocities of all Nosé-Hoover particles in the chain to the `.edr` file. Leap-frog simulations currently can only have Nosé-Hoover chain lengths of 1, but this will likely be updated in later version.

As described in the integrator section, for temperature coupling, the temperature that the algorithm attempts to match to the reference temperature is calculated differently in velocity Verlet and leap-frog dynamics. Velocity Verlet (*md-vv*) uses the full-step kinetic energy, while leap-frog and *md-vv-avek* use the half-step-averaged kinetic energy.

We can examine the Trotter decomposition again to better understand the differences between these constant-temperature integrators. In the case of Nosé-Hoover dynamics (for simplicity, using a chain with $N = 1$, with more details in Ref. [34]), we split the Liouville operator as

$$iL = iL_1 + iL_2 + iL_{\mathrm{NHC}}, \tag{3.54}$$

where

$$iL_1 = \sum_{i=1}^{N}\left[\frac{\boldsymbol{p}_i}{m_i}\right]\cdot\frac{\partial}{\partial\boldsymbol{r}_i}$$

$$\begin{aligned} iL_2 &= \sum_{i=1}^{N} \boldsymbol{F}_i \cdot \frac{\partial}{\partial \boldsymbol{p}_i} \\ iL_{\rm NHC} &= \sum_{i=1}^{N} -\frac{p_\xi}{Q} \boldsymbol{v}_i \cdot \nabla_{\boldsymbol{v}_i} + \frac{p_\xi}{Q}\frac{\partial}{\partial \xi} + (T - T_0)\frac{\partial}{\partial p_\xi} \end{aligned} \tag{3.55}$$

For standard velocity Verlet with Nosé-Hoover temperature control, this becomes

$$\begin{aligned} \exp(iL\Delta t) &= \exp\left(iL_{\rm NHC}\Delta t/2\right)\exp\left(iL_2\Delta t/2\right) \\ & \exp\left(iL_1\Delta t\right)\exp\left(iL_2\Delta t/2\right)\exp\left(iL_{\rm NHC}\Delta t/2\right) + \mathcal{O}(\Delta t^3). \end{aligned} \tag{3.56}$$

For half-step-averaged temperature control using *md-vv-avek*, this decomposition will not work, since we do not have the full step temperature until after the second velocity step. However, we can construct an alternate decomposition that is still reversible, by switching the place of the NHC and velocity portions of the decomposition:

$$\begin{aligned} \exp(iL\Delta t) &= \exp\left(iL_2\Delta t/2\right)\exp\left(iL_{\rm NHC}\Delta t/2\right)\exp\left(iL_1\Delta t\right) \\ & \exp\left(iL_{\rm NHC}\Delta t/2\right)\exp\left(iL_2\Delta t/2\right) + \mathcal{O}(\Delta t^3) \end{aligned} \tag{3.57}$$

This formalism allows us to easily see the difference between the different flavors of velocity Verlet integrator. The leap-frog integrator can be seen as starting with Eq. 3.57 just before the $\exp\left(iL_1\Delta t\right)$ term, yielding:

$$\begin{aligned} \exp(iL\Delta t) &= \exp\left(iL_1\Delta t\right)\exp\left(iL_{\rm NHC}\Delta t/2\right) \\ & \exp\left(iL_2\Delta t\right)\exp\left(iL_{\rm NHC}\Delta t/2\right) + \mathcal{O}(\Delta t^3) \end{aligned} \tag{3.58}$$

and then using some algebra tricks to solve for some quantities are required before they are actually calculated [35].

Group temperature coupling

In GROMACS temperature coupling can be performed on groups of atoms, typically a protein and solvent. The reason such algorithms were introduced is that energy exchange between different components is not perfect, due to different effects including cut-offs etc. If now the whole system is coupled to one heat bath, water (which experiences the largest cut-off noise) will tend to heat up and the protein will cool down. Typically 100 K differences can be obtained. With the use of proper electrostatic methods (PME) these difference are much smaller but still not negligible. The parameters for temperature coupling in groups are given in the `mdp` file. Recent investigation has shown that small temperature differences between protein and water may actually be an artifact of the way temperature is calculated when there are finite timesteps, and very large differences in temperature are likely a sign of something else seriously going wrong with the system, and should be investigated carefully [36].

One special case should be mentioned: it is possible to temperature-couple only part of the system, leaving other parts without temperature coupling. This is done by specifying -1 for the time constant τ_T for the group that should not be thermostatted. If only part of the system is thermostatted, the system will still eventually converge to an NVT system. In fact, one suggestion for minimizing errors in the temperature caused by discretized timesteps is that if constraints on the water are used, then only the water degrees of freedom should be thermostatted, not protein degrees of freedom, as the higher frequency modes in the protein can cause larger deviations from the "true" temperature, the temperature obtained with small timesteps [36].

3.4.9 Pressure coupling

In the same spirit as the temperature coupling, the system can also be coupled to a "pressure bath." GROMACS supports both the Berendsen algorithm [25] that scales coordinates and box vectors every step, the extended-ensemble Parrinello-Rahman approach [37, 38], and for the velocity Verlet variants, the Martyna-Tuckerman-Tobias-Klein (MTTK) implementation of pressure control [34]. Parrinello-Rahman and Berendsen can be combined with any of the temperature coupling methods above; MTTK can only be used with Nosé-Hoover temperature control.

Berendsen pressure coupling

The Berendsen algorithm rescales the coordinates and box vectors every step, or every $n_{\rm PC}$ steps, with a matrix $\boldsymbol{\mu}$, which has the effect of a first-order kinetic relaxation of the pressure towards a given reference pressure $\mathbf{P}_0$ according to

$$\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} = \frac{\mathbf{P}_0 - \mathbf{P}}{\tau_p}. \tag{3.59}$$

The scaling matrix $\boldsymbol{\mu}$ is given by

$$\mu_{ij} = \delta_{ij} - \frac{n_{\rm PC}\Delta t}{3\,\tau_p}\beta_{ij}\{P_{0ij} - P_{ij}(t)\}. \tag{3.60}$$

Here, $\boldsymbol{\beta}$ is the isothermal compressibility of the system. In most cases this will be a diagonal matrix, with equal elements on the diagonal, the value of which is generally not known. It suffices to take a rough estimate because the value of $\boldsymbol{\beta}$ only influences the non-critical time constant of the pressure relaxation without affecting the average pressure itself. For water at 1 atm and 300 K $\beta = 4.6 \times 10^{-10}$ Pa^{-1} $= 4.6 \times 10^{-5}$ bar^{-1}, which is 7.6×10^{-4} MD units (see chapter 2). Most other liquids have similar values. When scaling completely anisotropically, the system has to be rotated in order to obey eqn. 3.1. This rotation is approximated in first order in the scaling, which is usually less than 10^{-4}. The actual scaling matrix $\boldsymbol{\mu}'$ is

$$\boldsymbol{\mu}' = \begin{pmatrix} \mu_{xx} & \mu_{xy}+\mu_{yx} & \mu_{xz}+\mu_{zx} \\ 0 & \mu_{yy} & \mu_{yz}+\mu_{zy} \\ 0 & 0 & \mu_{zz} \end{pmatrix}. \tag{3.61}$$

The velocities are neither scaled nor rotated.

In GROMACS, the Berendsen scaling can also be done isotropically, which means that instead of $\boldsymbol{P}$ a diagonal matrix with elements of size trace$(\boldsymbol{P})/3$ is used. For systems with interfaces, semi-isotropic scaling can be useful. In this case, the x/y-directions are scaled isotropically and the z direction is scaled independently. The compressibility in the x/y or z-direction can be set to zero, to scale only in the other direction(s).

If you allow full anisotropic deformations and use constraints you might have to scale more slowly or decrease your timestep to avoid errors from the constraint algorithms. It is important to note that although the Berendsen pressure control algorithm yields a simulation with the correct average pressure, it does not yield the exact NPT ensemble, and it is not yet clear exactly what errors this approximation may yield.

Parrinello-Rahman pressure coupling

In cases where the fluctuations in pressure or volume are important *per se* (*e.g.* to calculate thermodynamic properties), especially for small systems, it may be a problem that the exact ensemble is not well defined for the weak-coupling scheme, and that it does not simulate the true NPT ensemble.

GROMACS also supports constant-pressure simulations using the Parrinello-Rahman approach [37, 38], which is similar to the Nosé-Hoover temperature coupling, and in theory gives the true NPT ensemble. With the Parrinello-Rahman barostat, the box vectors as represented by the matrix $\boldsymbol{b}$ obey the matrix equation of motion[2]

$$\frac{\mathrm{d}\boldsymbol{b}^2}{\mathrm{d}t^2} = V\boldsymbol{W}^{-1}\boldsymbol{b}'^{-1}\left(\boldsymbol{P} - \boldsymbol{P}_{ref}\right). \tag{3.62}$$

The volume of the box is denoted V, and $\boldsymbol{W}$ is a matrix parameter that determines the strength of the coupling. The matrices $\boldsymbol{P}$ and $\boldsymbol{P}_{ref}$ are the current and reference pressures, respectively.

The equations of motion for the particles are also changed, just as for the Nosé-Hoover coupling. In most cases you would combine the Parrinello-Rahman barostat with the Nosé-Hoover thermostat, but to keep it simple we only show the Parrinello-Rahman modification here:

$$\frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} = \frac{\boldsymbol{F}_i}{m_i} - \boldsymbol{M}\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t}, \tag{3.63}$$

$$\boldsymbol{M} = \boldsymbol{b}^{-1}\left[\boldsymbol{b}\frac{\mathrm{d}\boldsymbol{b}'}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}t}\boldsymbol{b}'\right]\boldsymbol{b}'^{-1}. \tag{3.64}$$

The (inverse) mass parameter matrix $\boldsymbol{W}^{-1}$ determines the strength of the coupling, and how the box can be deformed. The box restriction (3.1) will be fulfilled automatically if the corresponding elements of $\boldsymbol{W}^{-1}$ are zero. Since the coupling strength also depends on the size of your box, we prefer to calculate it automatically in GROMACS. You only have to provide the approximate isothermal compressibilities $\boldsymbol{\beta}$ and the pressure time constant τ_p in the input file (L is the largest box matrix element):

$$\left(\boldsymbol{W}^{-1}\right)_{ij} = \frac{4\pi^2\beta_{ij}}{3\tau_p^2 L}. \tag{3.65}$$

Just as for the Nosé-Hoover thermostat, you should realize that the Parrinello-Rahman time constant is *not* equivalent to the relaxation time used in the Berendsen pressure coupling algorithm.

[2]The box matrix representation $\boldsymbol{b}$ in GROMACS corresponds to the transpose of the box matrix representation $\boldsymbol{h}$ in the paper by Nosé and Klein. Because of this, some of our equations will look slightly different.

In most cases you will need to use a 4–5 times larger time constant with Parrinello-Rahman coupling. If your pressure is very far from equilibrium, the Parrinello-Rahman coupling may result in very large box oscillations that could even crash your run. In that case you would have to increase the time constant, or (better) use the weak-coupling scheme to reach the target pressure, and then switch to Parrinello-Rahman coupling once the system is in equilibrium. Additionally, using the leap-frog algorithm, the pressure at time t is not available until after the time step has completed, and so the pressure from the previous step must be used, which makes the algorithm not directly reversible, and may not be appropriate for high precision thermodynamic calculations.

Surface-tension coupling

When a periodic system consists of more than one phase, separated by surfaces which are parallel to the xy-plane, the surface tension and the z-component of the pressure can be coupled to a pressure bath. Presently, this only works with the Berendsen pressure coupling algorithm in GROMACS. The average surface tension $\gamma(t)$ can be calculated from the difference between the normal and the lateral pressure

$$\begin{aligned}\gamma(t) &= \frac{1}{n}\int_0^{L_z}\left\{P_{zz}(z,t)-\frac{P_{xx}(z,t)+P_{yy}(z,t)}{2}\right\}\mathrm{d}z && (3.66)\\ &= \frac{L_z}{n}\left\{P_{zz}(t)-\frac{P_{xx}(t)+P_{yy}(t)}{2}\right\}, && (3.67)\end{aligned}$$

where L_z is the height of the box and n is the number of surfaces. The pressure in the z-direction is corrected by scaling the height of the box with μ_{zz}

$$\Delta P_{zz} = \frac{\Delta t}{\tau_p}\{P_{0zz} - P_{zz}(t)\} \tag{3.68}$$

$$\mu_{zz} = 1 + \beta_{zz}\Delta P_{zz} \tag{3.69}$$

This is similar to normal pressure coupling, except that the factor of $1/3$ is missing. The pressure correction in the z-direction is then used to get the correct convergence for the surface tension to the reference value γ_0. The correction factor for the box length in the x/y-direction is

$$\mu_{x/y} = 1 + \frac{\Delta t}{2\,\tau_p}\beta_{x/y}\left(\frac{n\gamma_0}{\mu_{zz}L_z} - \left\{P_{zz}(t)+\Delta P_{zz} - \frac{P_{xx}(t)+P_{yy}(t)}{2}\right\}\right) \tag{3.70}$$

The value of β_{zz} is more critical than with normal pressure coupling. Normally an incorrect compressibility will just scale τ_p, but with surface tension coupling it affects the convergence of the surface tension. When β_{zz} is set to zero (constant box height), ΔP_{zz} is also set to zero, which is necessary for obtaining the correct surface tension.

MTTK pressure control algorithms

As mentioned in the previous section, one weakness of leap-frog integration is in constant pressure simulations, since the pressure requires a calculation of both the virial and the kinetic energy at the full time step; for leap-frog, this information is not available until *after* the full timestep. Velocity

The next step is to add temperature control. Adding Nosé-Hoover chains, including to the barostat degree of freedom, where we use η for the barostat Nosé-Hoover variables, and Q' for the coupling constants of the thermostats of the barostats, we get

$$\begin{aligned}
\dot{\boldsymbol{r}}_i &= \frac{\boldsymbol{p}_i}{m_i} + \frac{p_\epsilon}{W}\boldsymbol{r}_i \\
\frac{\dot{\boldsymbol{p}}_i}{m_i} &= \frac{1}{m_i}\boldsymbol{F}_i - \alpha\frac{p_\epsilon}{W}\frac{\boldsymbol{p}_i}{m_i} - \frac{p_{\xi_1}}{Q_1}\frac{\boldsymbol{p}_i}{m_i} \\
\dot{\epsilon} &= \frac{p_\epsilon}{W} \\
\frac{\dot{p}_\epsilon}{W} &= \frac{3V}{W}(\alpha P_{\text{kin}} - P_{\text{vir}} - P) - \frac{p_{\eta_1}}{Q'_1}p_\epsilon \\
\dot{\xi}_k &= \frac{p_{\xi_k}}{Q_k} \\
\dot{\eta}_k &= \frac{p_{\eta_k}}{Q'_k} \\
\dot{p}_{\xi_k} &= G_k - \frac{p_{\xi_{k+1}}}{Q_{k+1}} \quad k = 1, \ldots, M-1 \\
\dot{p}_{\eta_k} &= G'_k - \frac{p_{\eta_{k+1}}}{Q'_{k+1}} \quad k = 1, \ldots, M-1 \\
\dot{p}_{\xi_M} &= G_M \\
\dot{p}_{\eta_M} &= G'_M,
\end{aligned} \tag{3.77}$$

where

$$\begin{aligned}
P_{\text{int}} &= P_{\text{kin}} - P_{\text{vir}} = \frac{1}{3V}\left[\sum_{i=1}^{N}\left(\frac{\boldsymbol{p}_i^2}{2m_i} - \boldsymbol{r}_i \cdot \boldsymbol{F}_i\right)\right] \\
G_1 &= \sum_{i=1}^{N}\frac{\boldsymbol{p}_i^2}{m_i} - N_f kT \\
G_k &= \frac{p_{\xi_{k-1}}^2}{2Q_{k-1}} - kT \quad k = 2, \ldots, M \\
G'_1 &= \frac{p_\epsilon^2}{2W} - kT \\
G'_k &= \frac{p_{\eta_{k-1}}^2}{2Q'_{k-1}} - kT \quad k = 2, \ldots, M
\end{aligned} \tag{3.78}$$

The conserved quantity is now

$$\begin{aligned}
H = \sum_{i=1}^{N}\frac{\boldsymbol{p}_i}{2m_i} + U(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) + \frac{p_\epsilon^2}{2W} + PV + \\
\sum_{k=1}^{M}\frac{p_{\xi_k}^2}{2Q_k} + \sum_{k=1}^{M}\frac{p_{\eta_k}^2}{2Q'_k} + N_f kT\xi_1 + kT\sum_{i=2}^{M}\xi_k + kT\sum_{k=1}^{M}\eta_k
\end{aligned} \tag{3.79}$$

Returning to the Trotter decomposition formalism, for pressure control and temperature control [34] we get:

$$iL = iL_1 + iL_2 + iL_{\epsilon,1} + iL_{\epsilon,2} + iL_{\mathrm{NHC-baro}} + iL_{\mathrm{NHC}} \tag{3.80}$$

where "NHC-baro" corresponds to the Nosè-Hoover chain of the barostat, and NHC corresponds to the NHC of the particles,

$$\begin{aligned}
iL_1 &= \sum_{i=1}^{N} \left[\frac{\boldsymbol{p}_i}{m_i} + \frac{p_\epsilon}{W}\boldsymbol{r}_i\right] \cdot \frac{\partial}{\partial \boldsymbol{r}_i} && (3.81)\\
iL_2 &= \sum_{i=1}^{N} \boldsymbol{F}_i - \alpha \frac{p_\epsilon}{W}\boldsymbol{p}_i \cdot \frac{\partial}{\partial \boldsymbol{p}_i} && (3.82)\\
iL_{\epsilon,1} &= \frac{p_\epsilon}{W}\frac{\partial}{\partial \epsilon} && (3.83)\\
iL_{\epsilon,2} &= G_\epsilon \frac{\partial}{\partial p_\epsilon} && (3.84)
\end{aligned}$$

and where

$$G_\epsilon = 3V\left(\alpha P_{\mathrm{kin}} - P_{\mathrm{vir}} - P\right) \tag{3.85}$$

Using the Trotter decomposition, we get

$$\begin{aligned}
\exp(iL\Delta t) = \; & \exp\left(iL_{\mathrm{NHC-baro}}\Delta t/2\right)\exp\left(iL_{\mathrm{NHC}}\Delta t/2\right)\\
& \exp\left(iL_{\epsilon,2}\Delta t/2\right)\exp\left(iL_2\Delta t/2\right)\\
& \exp\left(iL_{\epsilon,1}\Delta t\right)\exp\left(iL_1\Delta t\right)\\
& \exp\left(iL_2\Delta t/2\right)\exp\left(iL_{\epsilon,2}\Delta t/2\right)\\
& \exp\left(iL_{\mathrm{NHC}}\Delta t/2\right)\exp\left(iL_{\mathrm{NHC-baro}}\Delta t/2\right) + \mathcal{O}(\Delta t^3)
\end{aligned} \tag{3.86}$$

The action of $\exp\left(iL_1\Delta t\right)$ comes from the solution of the the differential equation $\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i + v_\epsilon \boldsymbol{r}_i$ with $\boldsymbol{v}_i = \boldsymbol{p}_i/m_i$ and v_ϵ constant with initial condition $\boldsymbol{r}_i(0)$, evaluate at $t = \Delta t$. This yields the evolution

$$\boldsymbol{r}_i(\Delta t) = \boldsymbol{r}_i(0)e^{v_\epsilon \Delta t} + \Delta t \boldsymbol{v}_i(0)e^{v_\epsilon \Delta t/2}\frac{\sinh\left(v_\epsilon \Delta t/2\right)}{v_\epsilon \Delta t/2}. \tag{3.87}$$

The action of $\exp\left(iL_2\Delta t/2\right)$ comes from the solution of the differential equation $\dot{\boldsymbol{v}}_i = \frac{\boldsymbol{F}_i}{m_i} - \alpha v_\epsilon \boldsymbol{v}_i$, yielding

$$\boldsymbol{v}_i(\Delta t/2) = \boldsymbol{v}_i(0)e^{-\alpha v_\epsilon \Delta t/2} + \frac{\Delta t}{2m_i}\boldsymbol{F}_i(0)e^{-\alpha v_\epsilon \Delta t/4}\frac{\sinh\left(\alpha v_\epsilon \Delta t/4\right)}{\alpha v_\epsilon \Delta t/4}. \tag{3.88}$$

md-vv-avek uses the full step kinetic energies for determining the pressure with the pressure control, but the half-step-averaged kinetic energy for the temperatures, which can be written as a Trotter decomposition as

$$\begin{aligned}
\exp(iL\Delta t) = \; & \exp\left(iL_{\mathrm{NHC-baro}}\Delta t/2\right)\exp\left(iL_{\epsilon,2}\Delta t/2\right)\exp\left(iL_2\Delta t/2\right)\\
& \exp\left(iL_{\mathrm{NHC}}\Delta t/2\right)\exp\left(iL_{\epsilon,1}\Delta t\right)\exp\left(iL_1\Delta t\right)\exp\left(iL_{\mathrm{NHC}}\Delta t/2\right)\\
& \exp\left(iL_2\Delta t/2\right)\exp\left(iL_{\epsilon,2}\Delta t/2\right)\exp\left(iL_{\mathrm{NHC-baro}}\Delta t/2\right) + \mathcal{O}(\Delta t^3)
\end{aligned} \tag{3.89}$$

With constraints, the equations become significantly more complicated, in that each of these equations need to be solved iteratively for the constraint forces. The discussion of the details of the iteration is beyond the scope of this manual; readers are encouraged to see the implementation described in [40].

Infrequent evaluation of temperature and pressure coupling

Temperature and pressure control require global communication to compute the kinetic energy and virial, which can become costly if performed every step for large systems. We can rearrange the Trotter decomposition to give alternate symplectic, reversible integrator with the coupling steps every n steps instead of every steps. These new integrators will diverge if the coupling time step is too large, as the auxiliary variable integrations will not converge. However, in most cases, long coupling times are more appropriate, as they disturb the dynamics less [34].

Standard velocity Verlet with Nosé-Hoover temperature control has a Trotter expansion

$$\begin{aligned}\exp(iL\Delta t) &\approx \exp\left(iL_{\mathrm{NHC}}\Delta t/2\right)\exp\left(iL_2\Delta t/2\right)\\ &\quad \exp\left(iL_1\Delta t\right)\exp\left(iL_2\Delta t/2\right)\exp\left(iL_{\mathrm{NHC}}\Delta t/2\right). \end{aligned} \tag{3.90}$$

If the Nosé-Hoover chain is sufficiently slow with respect to the motions of the system, we can write an alternate integrator over n steps for velocity Verlet as

$$\begin{aligned}\exp(iL\Delta t) &\approx \left(\exp\left(iL_{\mathrm{NHC}}(n\Delta t/2)\right)\left[\exp\left(iL_2\Delta t/2\right)\right.\right.\\ &\quad \left.\left.\exp\left(iL_1\Delta t\right)\exp\left(iL_2\Delta t/2\right)\right]^n\exp\left(iL_{\mathrm{NHC}}(n\Delta t/2)\right)\right). \end{aligned} \tag{3.91}$$

For pressure control, this becomes

$$\begin{aligned}\exp(iL\Delta t) &\approx \exp\left(iL_{\mathrm{NHC-baro}}(n\Delta t/2)\right)\exp\left(iL_{\mathrm{NHC}}(n\Delta t/2)\right)\\ &\quad \exp\left(iL_{\epsilon,2}(n\Delta t/2)\right)\left[\exp\left(iL_2\Delta t/2\right)\right.\\ &\quad \exp\left(iL_{\epsilon,1}\Delta t\right)\exp\left(iL_1\Delta t\right)\\ &\quad \left.\exp\left(iL_2\Delta t/2\right)\right]^n\exp\left(iL_{\epsilon,2}(n\Delta t/2)\right)\\ &\quad \exp\left(iL_{\mathrm{NHC}}(n\Delta t/2)\right)\exp\left(iL_{\mathrm{NHC-baro}}(n\Delta t/2)\right), \end{aligned} \tag{3.92}$$

where the box volume integration occurs every step, but the auxiliary variable integrations happen every n steps.

3.4.10 The complete update algorithm

The complete algorithm for the update of velocities and coordinates is given using leap-frog in Fig. 3.9. The SHAKE algorithm of step 4 is explained below.

GROMACS has a provision to "freeze" (prevent motion of) selected particles, which must be defined as a "freeze group." This is implemented using a *freeze factor* $\boldsymbol{f}_g$, which is a vector, and differs for each freeze group (see sec. 3.3). This vector contains only zero (freeze) or one (don't freeze). When we take this freeze factor and the external acceleration $\boldsymbol{a}_h$ into account the update algorithm for the velocities becomes

$$\boldsymbol{v}(t+\frac{\Delta t}{2}) \;=\; \boldsymbol{f}_g * \lambda * \left[\boldsymbol{v}(t-\frac{\Delta t}{2})+\frac{\boldsymbol{F}(t)}{m}\Delta t+\boldsymbol{a}_h\Delta t\right], \tag{3.93}$$

THE UPDATE ALGORITHM

Given:
Positions $\boldsymbol{r}$ of all atoms at time t
Velocities $\boldsymbol{v}$ of all atoms at time $t - \frac{1}{2}\Delta t$
Accelerations $\boldsymbol{F}/m$ on all atoms at time t.
(Forces are computed disregarding any constraints)
Total kinetic energy and virial at $t - \Delta t$

$\Downarrow$

1. Compute the scaling factors λ and μ
according to eqns. 3.45 and 3.60

$\Downarrow$

2. Update and scale velocities: $\boldsymbol{v}' = \lambda(\boldsymbol{v} + \boldsymbol{a}\Delta t)$

$\Downarrow$

3. Compute new unconstrained coordinates: $\boldsymbol{r}' = \boldsymbol{r} + \boldsymbol{v}'\Delta t$

$\Downarrow$

4. Apply constraint algorithm to coordinates: constrain($\boldsymbol{r}' \rightarrow \boldsymbol{r}''$; $\boldsymbol{r}$)

$\Downarrow$

5. Correct velocities for constraints: $\boldsymbol{v} = (\boldsymbol{r}'' - \boldsymbol{r})/\Delta t$

$\Downarrow$

6. Scale coordinates and box: $\boldsymbol{r} = \mu\boldsymbol{r}''; \boldsymbol{b} = \mu\boldsymbol{b}$

Figure 3.9: The MD update algorithm with the leap-frog integrator

where g and h are group indices which differ per atom.

3.4.11 Output step

The most important output of the MD run is the *trajectory file*, which contains particle coordinates and (optionally) velocities at regular intervals. The trajectory file contains frames that could include positions, velocities and/or forces, as well as information about the dimensions of the simulation volume, integration step, integration time, etc. The interpretation of the time varies with the integrator chosen, as described above. For Velocity Verlet integrators, velocities labeled at time t are for that time. For other integrators (e.g. leap-frog, stochastic dynamics), the velocities labeled at time t are for time $t - \frac{1}{2}\Delta t$.

Since the trajectory files are lengthy, one should not save every step! To retain all information it suffices to write a frame every 15 steps, since at least 30 steps are made per period of the highest frequency in the system, and Shannon's sampling theorem states that two samples per period of the highest frequency in a band-limited signal contain all available information. But that still gives very long files! So, if the highest frequencies are not of interest, 10 or 20 samples per ps may suffice. Be aware of the distortion of high-frequency motions by the *stroboscopic effect*, called *aliasing*: higher frequencies are mirrored with respect to the sampling frequency and appear as lower frequencies.

GROMACS can also write reduced-precision coordinates for a subset of the simulation system to a special compressed trajectory file format. All the other tools can read and write this format. See sec. 7.3 for details on how to set up your `.mdp` file to have `mdrun` use this feature.