

4.6 Methods

4.6.1 Exclusions and 1-4 Interactions.

Atoms within a molecule that are close by in the chain, *i.e.* atoms that are covalently bonded, or linked by one or two atoms are called *first neighbors*, *second neighbors* and *third neighbors*, respectively (see Fig. 4.18). Since the interactions of atom **i** with atoms **i+1** and **i+2** are mainly quantum mechanical, they can not be modeled by a Lennard-Jones potential. Instead it is assumed that these interactions are adequately modeled by a harmonic bond term or constraint (**i**, **i+1**) and a harmonic angle term (**i**, **i+2**). The first and second neighbors (atoms **i+1** and **i+2**) are therefore *excluded* from the Lennard-Jones interaction list of atom **i**; atoms **i+1** and **i+2** are called *exclusions* of atom **i**.

For third neighbors, the normal Lennard-Jones repulsion is sometimes still too strong, which means that when applied to a molecule, the molecule would deform or break due to the internal strain. This is especially the case for carbon-carbon interactions in a *cis*-conformation (*e.g.* *cis*-butane). Therefore, for some of these interactions, the Lennard-Jones repulsion has been reduced in the GROMOS force field, which is implemented by keeping a separate list of 1-4 and normal Lennard-Jones parameters. In other force fields, such as OPLS [101], the standard Lennard-

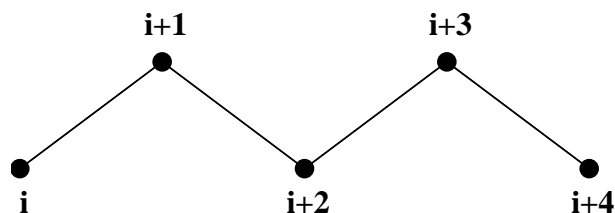


Figure 4.18: Atoms along an alkane chain.

Jones parameters are reduced by a factor of two, but in that case also the dispersion (r^{-6}) and the Coulomb interaction are scaled. GROMACS can use either of these methods.

4.6.2 Charge Groups

In principle, the force calculation in MD is an $O(N^2)$ problem. Therefore, we apply a cut-off for non-bonded force (NBF) calculations; only the particles within a certain distance of each other are interacting. This reduces the cost to $O(N)$ (typically $100N$ to $200N$) of the NBF. It also introduces an error, which is, in most cases, acceptable, except when applying the cut-off implies the creation of charges, in which case you should consider using the lattice sum methods provided by GROMACS.

Consider a water molecule interacting with another atom. If we would apply a plain cut-off on an atom-atom basis we might include the atom-oxygen interaction (with a charge of -0.82) without the compensating charge of the protons, and as a result, induce a large dipole moment over the system. Therefore, we have to keep groups of atoms with total charge 0 together. These groups are called *charge groups*. Note that with a proper treatment of long-range electrostatics (e.g. particle-mesh Ewald (sec. 4.8.2), keeping charge groups together is not required.

4.6.3 Treatment of Cut-offs in the group scheme

GROMACS is quite flexible in treating cut-offs, which implies there can be quite a number of parameters to set. These parameters are set in the input file for `grompp`. There are two sort of parameters that affect the cut-off interactions; you can select which type of interaction to use in each case, and which cut-offs should be used in the neighbor searching.

For both Coulomb and van der Waals interactions there are interaction type selectors (termed `vdwtype` and `coulombtype`) and two parameters, for a total of six non-bonded interaction parameters. See sec. 7.3 for a complete description of these parameters.

The neighbor searching (NS) can be performed using a single-range, or a twin-range approach. Since the former is merely a special case of the latter, we will discuss the more general twin-range. In this case, NS is described by two radii: `rlist` and `max(rcoulomb,rvdw)`. Usually one builds the neighbor list every 10 time steps or every 20 fs (parameter `nstlist`). In the neighbor list, all interaction pairs that fall within `rlist` are stored. Furthermore, the interactions between pairs that do not fall within `rlist` but do fall within `max(rcoulomb,rvdw)` are computed during NS. The forces and energy are stored separately and added to short-range forces at every time step between successive NS. If `rlist` = `max(rcoulomb,rvdw)`, no forces are

evaluated during neighbor list generation. The virial is calculated from the sum of the short- and long-range forces. This means that the virial can be slightly asymmetrical at non-NS steps. When mdrun is compiled to use mixed precision, the virial is almost always asymmetrical because the off-diagonal elements are about as large as each element in the sum. In most cases this is not really a problem, since the fluctuations in the virial can be 2 orders of magnitude larger than the average.

Except for the plain cut-off, all of the interaction functions in Table 4.2 require that neighbor searching be done with a larger radius than the r_c specified for the functional form, because of the use of charge groups. The extra radius is typically of the order of 0.25 nm (roughly the largest distance between two atoms in a charge group plus the distance a charge group can diffuse within neighbor list updates).

	Type	Parameters
Coulomb	Plain cut-off	r_c, ϵ_r
	Reaction field	r_c, ϵ_{rf}
	Shift function	r_1, r_c, ϵ_r
	Switch function	r_1, r_c, ϵ_r
VdW	Plain cut-off	r_c
	Shift function	r_1, r_c
	Switch function	r_1, r_c

Table 4.2: Parameters for the different functional forms of the non-bonded interactions.

4.7 Virtual interaction sites

Virtual interaction sites (called dummy atoms in GROMACS versions before 3.3) can be used in GROMACS in a number of ways. We write the position of the virtual site \mathbf{r}_s as a function of the positions of other particles \mathbf{r}_i : $\mathbf{r}_s = f(\mathbf{r}_1.. \mathbf{r}_n)$. The virtual site, which may carry charge or be involved in other interactions, can now be used in the force calculation. The force acting on the virtual site must be redistributed over the particles with mass in a consistent way. A good way to do this can be found in ref. [102]. We can write the potential energy as:

$$V = V(\mathbf{r}_s, \mathbf{r}_1, \dots, \mathbf{r}_n) = V^*(\mathbf{r}_1, \dots, \mathbf{r}_n) \quad (4.141)$$

The force on the particle i is then:

$$\mathbf{F}_i = -\frac{\partial V^*}{\partial \mathbf{r}_i} = -\frac{\partial V}{\partial \mathbf{r}_i} - \frac{\partial V}{\partial \mathbf{r}_s} \frac{\partial \mathbf{r}_s}{\partial \mathbf{r}_i} = \mathbf{F}_i^{\text{direct}} + \mathbf{F}_i' \quad (4.142)$$

The first term is the normal force. The second term is the force on particle i due to the virtual site, which can be written in tensor notation:

$$\mathbf{F}_i' = \begin{bmatrix} \frac{\partial x_s}{\partial x_i} & \frac{\partial y_s}{\partial x_i} & \frac{\partial z_s}{\partial x_i} \\ \frac{\partial x_s}{\partial y_i} & \frac{\partial y_s}{\partial y_i} & \frac{\partial z_s}{\partial y_i} \\ \frac{\partial x_s}{\partial z_i} & \frac{\partial y_s}{\partial z_i} & \frac{\partial z_s}{\partial z_i} \end{bmatrix} \mathbf{F}_s \quad (4.143)$$

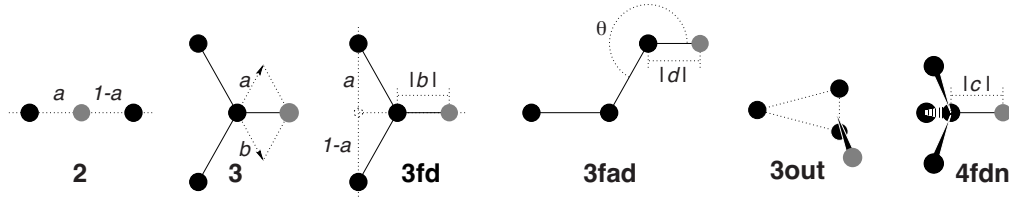


Figure 4.19: The six different types of virtual site construction in GROMACS. The constructing atoms are shown as black circles, the virtual sites in gray.

where \mathbf{F}_s is the force on the virtual site and x_s , y_s and z_s are the coordinates of the virtual site. In this way, the total force and the total torque are conserved [102].

The computation of the virial (eqn. 3.24) is non-trivial when virtual sites are used. Since the virial involves a summation over all the atoms (rather than virtual sites), the forces must be redistributed from the virtual sites to the atoms (using eqn. 4.143) *before* computation of the virial. In some special cases where the forces on the atoms can be written as a linear combination of the forces on the virtual sites (types 2 and 3 below) there is no difference between computing the virial before and after the redistribution of forces. However, in the general case redistribution should be done first.

There are six ways to construct virtual sites from surrounding atoms in GROMACS, which we classify by the number of constructing atoms. **Note** that all site types mentioned can be constructed from types 3fd (normalized, in-plane) and 3out (non-normalized, out of plane). However, the amount of computation involved increases sharply along this list, so we strongly recommended using the first adequate virtual site type that will be sufficient for a certain purpose. Fig. 4.19 depicts 6 of the available virtual site constructions. The conceptually simplest construction types are linear combinations:

$$\mathbf{r}_s = \sum_{i=1}^N w_i \mathbf{r}_i \quad (4.144)$$

The force is then redistributed using the same weights:

$$\mathbf{F}'_i = w_i \mathbf{F}_s \quad (4.145)$$

The types of virtual sites supported in GROMACS are given in the list below. Constructing atoms in virtual sites can be virtual sites themselves, but only if they are higher in the list, i.e. virtual sites can be constructed from “particles” that are simpler virtual sites.

2. As a linear combination of two atoms (Fig. 4.19 2):

$$w_i = 1 - a, \quad w_j = a \quad (4.146)$$

In this case the virtual site is on the line through atoms i and j .

3. As a linear combination of three atoms (Fig. 4.19 3):

$$w_i = 1 - a - b, \quad w_j = a, \quad w_k = b \quad (4.147)$$

In this case the virtual site is in the plane of the other three particles.

3fd. In the plane of three atoms, with a fixed distance (Fig. 4.19 3fd):

$$\mathbf{r}_s = \mathbf{r}_i + b \frac{\mathbf{r}_{ij} + a\mathbf{r}_{jk}}{|\mathbf{r}_{ij} + a\mathbf{r}_{jk}|} \quad (4.148)$$

In this case the virtual site is in the plane of the other three particles at a distance of $|b|$ from i . The force on particles i, j and k due to the force on the virtual site can be computed as:

$$\begin{aligned} \mathbf{F}'_i &= \mathbf{F}_s - \gamma(\mathbf{F}_s - \mathbf{p}) \\ \mathbf{F}'_j &= (1 - a)\gamma(\mathbf{F}_s - \mathbf{p}) \\ \mathbf{F}'_k &= a\gamma(\mathbf{F}_s - \mathbf{p}) \end{aligned} \quad \text{where} \quad \gamma = \frac{b}{|\mathbf{r}_{ij} + a\mathbf{r}_{jk}|} \quad \mathbf{p} = \frac{\mathbf{r}_{is} \cdot \mathbf{F}_s}{\mathbf{r}_{is} \cdot \mathbf{r}_{is}} \mathbf{r}_{is} \quad (4.149)$$

3fad. In the plane of three atoms, with a fixed angle and distance (Fig. 4.19 3fad):

$$\mathbf{r}_s = \mathbf{r}_i + d \cos \theta \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} + d \sin \theta \frac{\mathbf{r}_\perp}{|\mathbf{r}_\perp|} \quad \text{where} \quad \mathbf{r}_\perp = \mathbf{r}_{jk} - \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{r}_{ij} \quad (4.150)$$

In this case the virtual site is in the plane of the other three particles at a distance of $|d|$ from i at an angle of α with \mathbf{r}_{ij} . Atom k defines the plane and the direction of the angle. **Note** that in this case b and α must be specified, instead of a and b (see also sec. 5.2.2). The force on particles i, j and k due to the force on the virtual site can be computed as (with \mathbf{r}_\perp as defined in eqn. 4.150):

$$\begin{aligned} \mathbf{F}'_i &= \mathbf{F}_s - \frac{d \cos \theta}{|\mathbf{r}_{ij}|} \mathbf{F}_1 + \frac{d \sin \theta}{|\mathbf{r}_\perp|} \left(\frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{F}_2 + \mathbf{F}_3 \right) \\ \mathbf{F}'_j &= \frac{d \cos \theta}{|\mathbf{r}_{ij}|} \mathbf{F}_1 - \frac{d \sin \theta}{|\mathbf{r}_\perp|} \left(\mathbf{F}_2 + \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{F}_2 + \mathbf{F}_3 \right) \\ \mathbf{F}'_k &= \frac{d \sin \theta}{|\mathbf{r}_\perp|} \mathbf{F}_2 \end{aligned}$$

where $\mathbf{F}_1 = \mathbf{F}_s - \frac{\mathbf{r}_{ij} \cdot \mathbf{F}_s}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{r}_{ij}$, $\mathbf{F}_2 = \mathbf{F}_1 - \frac{\mathbf{r}_\perp \cdot \mathbf{F}_s}{\mathbf{r}_\perp \cdot \mathbf{r}_\perp} \mathbf{r}_\perp$ and $\mathbf{F}_3 = \frac{\mathbf{r}_{ij} \cdot \mathbf{F}_s}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{r}_\perp$ (4.151)

3out. As a non-linear combination of three atoms, out of plane (Fig. 4.19 3out):

$$\mathbf{r}_s = \mathbf{r}_i + a\mathbf{r}_{ij} + b\mathbf{r}_{ik} + c(\mathbf{r}_{ij} \times \mathbf{r}_{ik}) \quad (4.152)$$

This enables the construction of virtual sites out of the plane of the other atoms. The force on particles i, j and k due to the force on the virtual site can be computed as:

$$\begin{aligned} \mathbf{F}'_j &= \begin{bmatrix} a & -c z_{ik} & c y_{ik} \\ c z_{ik} & a & -c x_{ik} \\ -c y_{ik} & c x_{ik} & a \end{bmatrix} \mathbf{F}_s \\ \mathbf{F}'_k &= \begin{bmatrix} b & c z_{ij} & -c y_{ij} \\ -c z_{ij} & b & c x_{ij} \\ c y_{ij} & -c x_{ij} & b \end{bmatrix} \mathbf{F}_s \\ \mathbf{F}'_i &= \mathbf{F}_s - \mathbf{F}'_j - \mathbf{F}'_k \end{aligned} \quad (4.153)$$

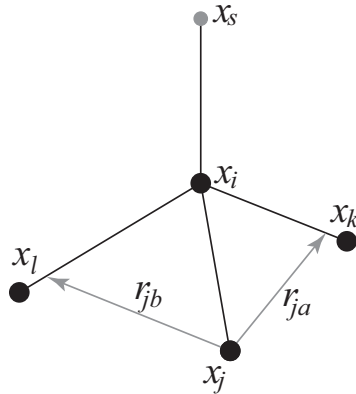


Figure 4.20: The new 4fdn virtual site construction, which is stable even when all constructing atoms are in the same plane.

4fdn. From four atoms, with a fixed distance, see separate Fig. 4.20. This construction is a bit complex, in particular since the previous type (4fd) could be unstable which forced us to introduce a more elaborate construction:

$$\begin{aligned}
 \mathbf{r}_{ja} &= a \mathbf{r}_{ik} - \mathbf{r}_{ij} = a (\mathbf{x}_k - \mathbf{x}_i) - (\mathbf{x}_j - \mathbf{x}_i) \\
 \mathbf{r}_{jb} &= b \mathbf{r}_{il} - \mathbf{r}_{ij} = b (\mathbf{x}_l - \mathbf{x}_i) - (\mathbf{x}_j - \mathbf{x}_i) \\
 \mathbf{r}_m &= \mathbf{r}_{ja} \times \mathbf{r}_{jb} \\
 \mathbf{x}_s &= \mathbf{x}_i + c \frac{\mathbf{r}_m}{|\mathbf{r}_m|}
 \end{aligned} \tag{4.154}$$

In this case the virtual site is at a distance of $|c|$ from i , while a and b are parameters. **Note** that the vectors \mathbf{r}_{ik} and \mathbf{r}_{il} are not normalized to save floating-point operations. The force on particles i , j , k and l due to the force on the virtual site are computed through chain rule derivatives of the construction expression. This is exact and conserves energy, but it does lead to relatively lengthy expressions that we do not include here (over 200 floating-point operations). The interested reader can look at the source code in `vsite.c`. Fortunately, this `vsite` type is normally only used for chiral centers such as C_α atoms in proteins.

The new 4fdn construct is identified with a ‘type’ value of 2 in the topology. The earlier 4fd type is still supported internally (‘type’ value 1), but it should not be used for new simulations. All current GROMACS tools will automatically generate type 4fdn instead.

N. A linear combination of N atoms with relative weights a_i . The weight for atom i is:

$$w_i = a_i \left(\sum_{j=1}^N a_j \right)^{-1} \tag{4.155}$$

There are three options for setting the weights:

COG center of geometry: equal weights

COM center of mass: a_i is the mass of atom i ; when in free-energy simulations the mass of the atom is changed, only the mass of the A-state is used for the weight

COW center of weights: a_i is defined by the user

4.8 Long Range Electrostatics

4.8.1 Ewald summation

The total electrostatic energy of N particles and their periodic images is given by

$$V = \frac{f}{2} \sum_{n_x} \sum_{n_y} \sum_{n_z^*} \sum_i^N \sum_j^N \frac{q_i q_j}{\mathbf{r}_{ij,\mathbf{n}}}. \quad (4.156)$$

$(n_x, n_y, n_z) = \mathbf{n}$ is the box index vector, and the star indicates that terms with $i = j$ should be omitted when $(n_x, n_y, n_z) = (0, 0, 0)$. The distance $\mathbf{r}_{ij,\mathbf{n}}$ is the real distance between the charges and not the minimum-image. This sum is conditionally convergent, but very slow.

Ewald summation was first introduced as a method to calculate long-range interactions of the periodic images in crystals [103]. The idea is to convert the single slowly-converging sum eqn. 4.156 into two quickly-converging terms and a constant term:

$$V = V_{\text{dir}} + V_{\text{rec}} + V_0 \quad (4.157)$$

$$V_{\text{dir}} = \frac{f}{2} \sum_{i,j}^N \sum_{n_x} \sum_{n_y} \sum_{n_z^*} q_i q_j \frac{\text{erfc}(\beta r_{ij,\mathbf{n}})}{r_{ij,\mathbf{n}}} \quad (4.158)$$

$$V_{\text{rec}} = \frac{f}{2\pi V} \sum_{i,j}^N q_i q_j \sum_{m_x} \sum_{m_y} \sum_{m_z^*} \frac{\exp(-(\pi \mathbf{m}/\beta)^2 + 2\pi i \mathbf{m} \cdot (\mathbf{r}_i - \mathbf{r}_j))}{\mathbf{m}^2} \quad (4.159)$$

$$V_0 = -\frac{f\beta}{\sqrt{\pi}} \sum_i^N q_i^2, \quad (4.160)$$

where β is a parameter that determines the relative weight of the direct and reciprocal sums and $\mathbf{m} = (m_x, m_y, m_z)$. In this way we can use a short cut-off (of the order of 1 nm) in the direct space sum and a short cut-off in the reciprocal space sum (e.g. 10 wave vectors in each direction). Unfortunately, the computational cost of the reciprocal part of the sum increases as N^2 (or $N^{3/2}$ with a slightly better algorithm) and it is therefore not realistic for use in large systems.

Using Ewald

Don't use Ewald unless you are absolutely sure this is what you want - for almost all cases the PME method below will perform much better. If you still want to employ classical Ewald summation enter this in your `.mdp` file, if the side of your box is about 3 nm:

```
coulombtype      = Ewald
rvdw             = 0.9
```

```
rlist           = 0.9
rcoulomb        = 0.9
fourierspacing  = 0.6
ewald-rtol      = 1e-5
```

The ratio of the box dimensions and the `fourierspacing` parameter determines the highest magnitude of wave vectors m_x, m_y, m_z to use in each direction. With a 3-nm cubic box this example would use 11 wave vectors (from -5 to 5) in each direction. The `ewald-rtol` parameter is the relative strength of the electrostatic interaction at the cut-off. Decreasing this gives you a more accurate direct sum, but a less accurate reciprocal sum.

4.8.2 PME

Particle-mesh Ewald is a method proposed by Tom Darden [12] to improve the performance of the reciprocal sum. Instead of directly summing wave vectors, the charges are assigned to a grid using interpolation. The implementation in GROMACS uses cardinal B-spline interpolation [13], which is referred to as smooth PME (SPME). The grid is then Fourier transformed with a 3D FFT algorithm and the reciprocal energy term obtained by a single sum over the grid in k-space.

The potential at the grid points is calculated by inverse transformation, and by using the interpolation factors we get the forces on each atom.

The PME algorithm scales as $N \log(N)$, and is substantially faster than ordinary Ewald summation on medium to large systems. On very small systems it might still be better to use Ewald to avoid the overhead in setting up grids and transforms. For the parallelization of PME see the section on MPMD PME (3.17.5).

With the Verlet cut-off scheme, the PME direct space potential is shifted by a constant such that the potential is zero at the cut-off. This shift is small and since the net system charge is close to zero, the total shift is very small, unlike in the case of the Lennard-Jones potential where all shifts add up. We apply the shift anyhow, such that the potential is the exact integral of the force.

Using PME

As an example for using Particle-mesh Ewald summation in GROMACS, specify the following lines in your `.mdp` file:

```
coulombtype     = PME
rvdw           = 0.9
rlist          = 0.9
rcoulomb       = 0.9
fourierspacing  = 0.12
pme-order      = 4
ewald-rtol     = 1e-5
```

In this case the `fourierspacing` parameter determines the maximum spacing for the FFT grid (i.e. minimum number of grid points), and `pme-order` controls the interpolation order. Using

fourth-order (cubic) interpolation and this spacing should give electrostatic energies accurate to about $5 \cdot 10^{-3}$. Since the Lennard-Jones energies are not this accurate it might even be possible to increase this spacing slightly.

Pressure scaling works with PME, but be aware of the fact that anisotropic scaling can introduce artificial ordering in some systems.

4.8.3 P3M-AD

The Particle-Particle Particle-Mesh methods of Hockney & Eastwood can also be applied in GROMACS for the treatment of long range electrostatic interactions [104]. Although the P3M method was the first efficient long-range electrostatics method for molecular simulation, the smooth PME (SPME) method has largely replaced P3M as the method of choice in atomistic simulations. One performance disadvantage of the original P3M method was that it required 3 3D-FFT back transforms to obtain the forces on the particles. But this is not required for P3M and the forces can be derived through analytical differentiation of the potential, as done in PME. The resulting method is termed P3M-AD. The only remaining difference between P3M-AD and PME is the optimization of the lattice Green influence function for error minimization that P3M uses. However, in 2012 it has been shown that the SPME influence function can be modified to obtain P3M [105]. This means that the advantage of error minimization in P3M-AD can be used at the same computational cost and with the same code as PME, just by adding a few lines to modify the influence function. However, at optimal parameter setting the effect of error minimization in P3M-AD is less than 10%. P3M-AD does show large accuracy gains with interlaced (also known as staggered) grids, but that is not supported in GROMACS (yet).

P3M is used in GROMACS with exactly the same options as used with PME by selecting the electrostatics type:

```
coulombtype      = P3M-AD
```

4.8.4 Optimizing Fourier transforms and PME calculations

It is recommended to optimize the parameters for calculation of electrostatic interaction such as PME grid dimensions and cut-off radii. This is particularly relevant to do before launching long production runs.

GROMACS includes a special tool, `g_tune_pme`, which automates the process of selecting the optimal size of the grid and number of PME-only nodes.

4.9 Long Range Van der Waals interactions

4.9.1 Dispersion correction

In this section, we derive long-range corrections due to the use of a cut-off for Lennard-Jones or Buckingham interactions. We assume that the cut-off is so long that the repulsion term can safely be neglected, and therefore only the dispersion term is taken into account. Due to the

nature of the dispersion interaction (we are truncating a potential proportional to $-r^{-6}$), energy and pressure corrections are both negative. While the energy correction is usually small, it may be important for free energy calculations where differences between two different Hamiltonians are considered. In contrast, the pressure correction is very large and can not be neglected under any circumstances where a correct pressure is required, especially for any NPT simulations. Although it is, in principle, possible to parameterize a force field such that the pressure is close to the desired experimental value without correction, such a method makes the parameterization dependent on the cut-off and is therefore undesirable.

Energy

The long-range contribution of the dispersion interaction to the virial can be derived analytically, if we assume a homogeneous system beyond the cut-off distance r_c . The dispersion energy between two particles is written as:

$$V(r_{ij}) = -C_6 r_{ij}^{-6} \quad (4.161)$$

and the corresponding force is:

$$\mathbf{F}_{ij} = -6 C_6 r_{ij}^{-8} \mathbf{r}_{ij} \quad (4.162)$$

In a periodic system it is not easy to calculate the full potentials, so usually a cut-off is applied, which can be abrupt or smooth. We will call the potential and force with cut-off V_c and \mathbf{F}_c . The long-range contribution to the dispersion energy in a system with N particles and particle density $\rho = N/V$ is:

$$V_{lr} = \frac{1}{2} N \rho \int_0^\infty 4\pi r^2 g(r) (V(r) - V_c(r)) dr \quad (4.163)$$

We will integrate this for the shift function, which is the most general form of van der Waals interaction available in GROMACS. The shift function has a constant difference S from 0 to r_1 and is 0 beyond the cut-off distance r_c . We can integrate eqn. 4.163, assuming that the density in the sphere within r_1 is equal to the global density and the radial distribution function $g(r)$ is 1 beyond r_1 :

$$\begin{aligned} V_{lr} &= \frac{1}{2} N \left(\rho \int_0^{r_1} 4\pi r^2 g(r) C_6 S dr + \rho \int_{r_1}^{r_c} 4\pi r^2 (V(r) - V_c(r)) dr + \rho \int_{r_c}^\infty 4\pi r^2 V(r) dr \right) \\ &= \frac{1}{2} N \left(\left(\frac{4}{3} \pi \rho r_1^3 - 1 \right) C_6 S + \rho \int_{r_1}^{r_c} 4\pi r^2 (V(r) - V_c(r)) dr - \frac{4}{3} \pi N \rho C_6 r_c^{-3} \right) \end{aligned} \quad (4.164)$$

where the term -1 corrects for the self-interaction. For a plain cut-off we only need to assume that $g(r)$ is 1 beyond r_c and the correction reduces to [106]:

$$V_{lr} = -\frac{2}{3} \pi N \rho C_6 r_c^{-3} \quad (4.165)$$

If we consider, for example, a box of pure water, simulated with a cut-off of 0.9 nm and a density of 1 g cm^{-3} this correction is $-0.75 \text{ kJ mol}^{-1}$ per molecule.

For a homogeneous mixture we need to define an *average dispersion constant*:

$$\langle C_6 \rangle = \frac{2}{N(N-1)} \sum_i^N \sum_{j>i}^N C_6(i, j) \quad (4.166)$$

In GROMACS, excluded pairs of atoms do not contribute to the average.

In the case of inhomogeneous simulation systems, *e.g.* a system with a lipid interface, the energy correction can be applied if $\langle C_6 \rangle$ for both components is comparable.

Virial and pressure

The scalar virial of the system due to the dispersion interaction between two particles i and j is given by:

$$\Xi = -\frac{1}{2} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} = 3 C_6 r_{ij}^{-6} \quad (4.167)$$

The pressure is given by:

$$P = \frac{2}{3V} (E_{kin} - \Xi) \quad (4.168)$$

The long-range correction to the virial is given by:

$$\Xi_{lr} = \frac{1}{2} N \rho \int_0^\infty 4\pi r^2 g(r) (\Xi - \Xi_c) dr \quad (4.169)$$

We can again integrate the long-range contribution to the virial assuming $g(r)$ is 1 beyond r_1 :

$$\begin{aligned} \Xi_{lr} &= \frac{1}{2} N \rho \left(\int_{r_1}^{r_c} 4\pi r^2 (\Xi - \Xi_c) dr + \int_{r_c}^\infty 4\pi r^2 3 C_6 r_{ij}^{-6} dr \right) \\ &= \frac{1}{2} N \rho \left(\int_{r_1}^{r_c} 4\pi r^2 (\Xi - \Xi_c) dr + 4\pi C_6 r_c^{-3} \right) \end{aligned} \quad (4.170)$$

For a plain cut-off the correction to the pressure is [106]:

$$P_{lr} = -\frac{4}{3} \pi C_6 \rho^2 r_c^{-3} \quad (4.171)$$

Using the same example of a water box, the correction to the virial is 0.75 kJ mol⁻¹ per molecule, the corresponding correction to the pressure for SPC water is approximately -280 bar.

For homogeneous mixtures, we can again use the average dispersion constant $\langle C_6 \rangle$ (eqn. 4.166):

$$P_{lr} = -\frac{4}{3} \pi \langle C_6 \rangle \rho^2 r_c^{-3} \quad (4.172)$$

For inhomogeneous systems, eqn. 4.172 can be applied under the same restriction as holds for the energy (see sec. 4.9.1).

4.9.2 Lennard-Jones PME

In order to treat systems, using Lennard-Jones potentials, that are non-homogeneous outside of the cut-off distance, we can instead use the Particle-mesh Ewald method as discussed for electrostatics above. In this case the modified Ewald equations become

$$V = V_{\text{dir}} + V_{\text{rec}} + V_0 \quad (4.173)$$

$$V_{\text{dir}} = -\frac{1}{2} \sum_{i,j} \sum_{n_x} \sum_{n_y} \sum_{n_z^*} \frac{C_6^{ij} g(\beta r_{ij,\mathbf{n}})}{r_{ij,\mathbf{n}}^6} \quad (4.174)$$

$$V_{\text{rec}} = \frac{\pi^{\frac{3}{2}} \beta^3}{2V} \sum_{m_x} \sum_{m_y} \sum_{m_z^*} f(\pi |\mathbf{m}|/\beta) \times \sum_{i,j} C_6^{ij} \exp[-2\pi i \mathbf{m} \cdot (\mathbf{r}_i - \mathbf{r}_j)] \quad (4.175)$$

$$V_0 = -\frac{\beta^6}{12} \sum_i C_6^{ii} \quad (4.176)$$

where $\mathbf{m} = (m_x, m_y, m_z)$, β is the parameter determining the weight between direct and reciprocal space, and C_6^{ij} is the combined dispersion parameter for particle i and j . The star indicates that terms with $i = j$ should be omitted when $((n_x, n_y, n_z) = (0, 0, 0))$, and $\mathbf{r}_{ij,\mathbf{n}}$ is the real distance between the particles. Following the derivation by Essmann [13], the functions f and g introduced above are defined as

$$f(x) = 1/3 \left[(1 - 2x^2) \exp(-x^2) + 2x^3 \sqrt{\pi} \operatorname{erfc}(x) \right] \quad (4.177)$$

$$g(x) = \exp(-x^2) \left(1 + x^2 + \frac{x^4}{2} \right). \quad (4.178)$$

The above methodology works fine as long as the dispersion parameters can be combined geometrically (eqn. 4.6) in the same way as the charges for electrostatics

$$C_{6,\text{geom}}^{ij} = \left(C_6^{ii} C_6^{jj} \right)^{1/2} \quad (4.179)$$

For Lorentz-Berthelot combination rules (eqn. 4.7), the reciprocal part of this sum has to be calculated seven times due to the splitting of the dispersion parameter according to

$$C_{6,\text{L-B}}^{ij} = (\sigma_i + \sigma_j)^6 = \sum_{n=0}^6 P_n \sigma_i^n \sigma_j^{(6-n)}, \quad (4.180)$$

for P_n the Pascal triangle coefficients. This introduces a non-negligible cost to the reciprocal part, requiring seven separate FFTs, and therefore this has been the limiting factor in previous attempts to implement LJ-PME. A solution to this problem is to use geometrical combination rules in order to calculate an approximate interaction parameter for the reciprocal part of the potential, yielding a total interaction of

$$\begin{aligned} V(r < r_c) &= \underbrace{C_6^{\text{dir}} g(\beta r) r^{-6}}_{\text{Direct space}} + \underbrace{C_{6,\text{geom}}^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{Reciprocal space}} \\ &= C_{6,\text{geom}}^{\text{recip}} r^{-6} + \left(C_6^{\text{dir}} - C_{6,\text{geom}}^{\text{recip}} \right) g(\beta r) r^{-6} \end{aligned} \quad (4.181)$$

$$V(r > r_c) = \underbrace{C_{6,\text{geom}}^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{Reciprocal space}}. \quad (4.182)$$

This will preserve a well-defined Hamiltonian and significantly increase the performance of the simulations. The approximation does introduce some errors, but since the difference is located in the interactions calculated in reciprocal space, the effect will be very small compared to the total

interaction energy. In a simulation of a lipid bilayer, using a cut-off of 1.0 nm, the relative error in total dispersion energy was below 0.5%. A more thorough discussion of this can be found in [107].

In GROMACS we now perform the proper calculation of this interaction by subtracting, from the direct-space interactions, the contribution made by the approximate potential that is used in the reciprocal part

$$V_{\text{dir}} = C_6^{\text{dir}} r^{-6} - C_6^{\text{recip}} [1 - g(\beta r)] r^{-6}. \quad (4.183)$$

This potential will reduce to the expression in eqn. 4.174 when $C_6^{\text{dir}} = C_6^{\text{recip}}$, and the total interaction is given by

$$\begin{aligned} V(r < r_c) &= \underbrace{C_6^{\text{dir}} r^{-6} - C_6^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{Direct space}} + \underbrace{C_6^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{Reciprocal space}} \\ &= C_6^{\text{dir}} r^{-6} \end{aligned} \quad (4.184)$$

$$V(r > r_c) = C_6^{\text{recip}} [1 - g(\beta r)] r^{-6}. \quad (4.185)$$

For the case when $C_6^{\text{dir}} \neq C_6^{\text{recip}}$ this will retain an unmodified LJ force up to the cut-off, and the error is an order of magnitude smaller than in simulations where the direct-space interactions do not account for the approximation used in reciprocal space. When using a VdW interaction modifier of potential-shift, the constant

$$\left(-C_6^{\text{dir}} + C_6^{\text{recip}} [1 - g(\beta r_c)] \right) r_c^{-6} \quad (4.186)$$

is added to eqn. 4.184 in order to ensure that the potential is continuous at the cutoff. Note that, in the same way as eqn. 4.183, this degenerates into the expected $-C_6 g(\beta r_c) r_c^{-6}$ when $C_6^{\text{dir}} = C_6^{\text{recip}}$. In addition to this, a long-range dispersion correction can be applied to correct for the approximation using a combination rule in reciprocal space. This correction assumes, as for the cut-off LJ potential, a uniform particle distribution. But since the error of the combination rule approximation is very small this long-range correction is not necessary in most cases. Also note that this homogenous correction does not correct the surface tension, which is an inhomogeneous property.

Using LJ-PME

As an example for using Particle-mesh Ewald summation for Lennard-Jones interactions in GROMACS, specify the following lines in your .mdp file:

```
vdwtype           = PME
rvdw              = 0.9
vdw-modifier      = Potential-Shift
rlist             = 0.9
rcoulomb          = 0.9
fourierspacing    = 0.12
pme-order         = 4
ewald-rtol-lj     = 0.001
lj-pme-comb-rule  = geometric
```

The same Fourier grid and interpolation order are used if both LJ-PME and electrostatic PME are active, so the settings for `fourierspacing` and `pme-order` are common to both. `ewald-rtol-lj` controls the splitting between direct and reciprocal space in the same way as `ewald-rtol`. In addition to this, the combination rule to be used in reciprocal space is determined by `lj-pme-comb-rule`. If the current force field uses Lorentz-Berthelot combination rules, it is possible to set `lj-pme-comb-rule = geometric` in order to gain a significant increase in performance for a small loss in accuracy. The details of this approximation can be found in the section above.

Note that the use of a complete long-range dispersion correction means that as with Coulomb PME, `rvdw` is now a free parameter in the method, rather than being necessarily restricted by the force-field parameterization scheme. Thus it is now possible to optimize the cutoff, spacing, order and tolerance terms for accuracy and best performance.

Naturally, the use of LJ-PME rather than LJ cut-off adds computation and communication done for the reciprocal-space part, so for best performance in balancing the load of parallel simulations using PME-only ranks, more such ranks should be used. It may be possible to improve upon the automatic load-balancing used by `mdrun`.

4.10 Force field

A force field is built up from two distinct components:

- The set of equations (called the *s*) used to generate the potential energies and their derivatives, the forces. These are described in detail in the previous chapter.
- The parameters used in this set of equations. These are not given in this manual, but in the data files corresponding to your GROMACS distribution.

Within one set of equations various sets of parameters can be used. Care must be taken that the combination of equations and parameters form a consistent set. It is in general dangerous to make *ad hoc* changes in a subset of parameters, because the various contributions to the total force are usually interdependent. This means in principle that every change should be documented, verified by comparison to experimental data and published in a peer-reviewed journal before it can be used.

GROMACS 5.0.2 includes several force fields, and additional ones are available on the website. If you do not know which one to select we recommend GROMOS-96 for united-atom setups and OPLS-AA/L for all-atom parameters. That said, we describe the available options in some detail.

All-hydrogen force field

The GROMOS-87-based all-hydrogen force field is almost identical to the normal GROMOS-87 force field, since the extra hydrogens have no Lennard-Jones interaction and zero charge. The only differences are in the bond angle and improper dihedral angle terms. This force field is only useful when you need the exact hydrogen positions, for instance for distance restraints derived from NMR measurements. When citing this force field please read the previous paragraph.

4.10.1 GROMOS-96

GROMACS supports the GROMOS-96 force fields [78]. All parameters for the 43A1, 43A2 (development, improved alkane dihedrals), 45A3, 53A5, and 53A6 parameter sets are included. All standard building blocks are included and topologies can be built automatically by `pdb2gmx`.

The GROMOS-96 force field is a further development of the GROMOS-87 force field. It has improvements over the GROMOS-87 force field for proteins and small molecules. **Note** that the sugar parameters present in 53A6 do correspond to those published in 2004[108], which are different from those present in 45A4, which is not included in GROMACS at this time. The 45A4 parameter set corresponds to a later revision of these parameters. The GROMOS-96 force field is not, however, recommended for use with long alkanes and lipids. The GROMOS-96 force field differs from the GROMOS-87 force field in a few respects:

- the force field parameters
- the parameters for the bonded interactions are not linked to atom types
- a fourth power bond stretching potential (4.2.1)
- an angle potential based on the cosine of the angle (4.2.6)

There are two differences in implementation between GROMACS and GROMOS-96 which can lead to slightly different results when simulating the same system with both packages:

- in GROMOS-96 neighbor searching for solvents is performed on the first atom of the solvent molecule. This is not implemented in GROMACS, but the difference with searching by centers of charge groups is very small
- the virial in GROMOS-96 is molecule-based. This is not implemented in GROMACS, which uses atomic virials

The GROMOS-96 force field was parameterized with a Lennard-Jones cut-off of 1.4 nm, so be sure to use a Lennard-Jones cut-off (`rvdw`) of at least 1.4. A larger cut-off is possible because the Lennard-Jones potential and forces are almost zero beyond 1.4 nm.

GROMOS-96 files

GROMACS can read and write GROMOS-96 coordinate and trajectory files. These files should have the extension `.g96`. Such a file can be a GROMOS-96 initial/final configuration file, a coordinate trajectory file, or a combination of both. The file is fixed format; all floats are written as 15.9, and as such, files can get huge. GROMACS supports the following data blocks in the given order:

- Header block:

```
TITLE (mandatory)
```

- Frame blocks:

```
TIMESTEP (optional)
POSITION/POSITIONRED (mandatory)
VELOCITY/VELOCITYRED (optional)
BOX (optional)
```

See the GROMOS-96 manual [78] for a complete description of the blocks. **Note** that all GROMACS programs can read compressed (.Z) or gzipped (.gz) files.

4.10.2 OPLS/AA

4.10.3 AMBER

GROMACS provides native support for the following AMBER force fields:

- AMBER94 [109]
- AMBER96 [110]
- AMBER99 [111]
- AMBER99SB [112]
- AMBER99SB-ILDN [113]
- AMBER03 [114]
- AMBERGS [115]

4.10.4 CHARMM

GROMACS supports the CHARMM force field for proteins [116, 117], lipids [118] and nucleic acids [119, 120]. The protein parameters (and to some extent the lipid and nucleic acid parameters) were thoroughly tested – both by comparing potential energies between the port and the standard parameter set in the CHARMM molecular simulation package, as well by how the protein force field behaves together with GROMACS-specific techniques such as virtual sites (enabling long time steps) and a fast implicit solvent recently implemented [73] – and the details and results are presented in the paper by Bjelkmar et al. [121]. The nucleic acid parameters, as well as the ones for HEME, were converted and tested by Michel Cuendet.

When selecting the CHARMM force field in `pdb2gmx` the default option is to use CMAP (for torsional correction map). To exclude CMAP, use `-nocmap`. The basic form of the CMAP term implemented in GROMACS is a function of the ϕ and ψ backbone torsion angles. This term is defined in the `.rtp` file by a `[cmap]` statement at the end of each residue supporting CMAP. The following five atom names define the two torsional angles. Atoms 1-4 define ϕ , and atoms 2-5 define ψ . The corresponding atom types are then matched to the correct CMAP type in the `cmap.itp` file that contains the correction maps.

A port of the CHARMM36 force field for use with GROMACS is also available at http://mackerell.umaryland.edu/charmm_ff.shtml#gromacs.

4.10.5 Coarse-grained force fields

Coarse-graining is a systematic way of reducing the number of degrees of freedom representing a system of interest. To achieve this, typically whole groups of atoms are represented by single beads and the coarse-grained force fields describes their effective interactions. Depending on the choice of parameterization, the functional form of such an interaction can be complicated and often tabulated potentials are used.

Coarse-grained models are designed to reproduce certain properties of a reference system. This can be either a full atomistic model or even experimental data. Depending on the properties to reproduce there are different methods to derive such force fields. An incomplete list of methods is given below:

- Conserving free energies
 - Simplex method
 - MARTINI force field (see next section)
- Conserving distributions (like the radial distribution function), so-called structure-based coarse-graining
 - (iterative) Boltzmann inversion
 - Inverse Monte Carlo
- Conserving forces
 - Force matching

Note that coarse-grained potentials are state dependent (e.g. temperature, density,...) and should be re-parametrized depending on the system of interest and the simulation conditions. This can for example be done using the Versatile Object-oriented Toolkit for Coarse-Graining Applications (VOTCA) [122]. The package was designed to assist in systematic coarse-graining, provides implementations for most of the algorithms mentioned above and has a well tested interface to GROMACS. It is available as open source and further information can be found at www.votca.org.

4.10.6 MARTINI

The MARTINI force field is a coarse-grain parameter set that allows for the construction of many systems, including proteins and membranes.

4.10.7 PLUM

The PLUM force field [123] is an example of a solvent-free protein-membrane model for which the membrane was derived from structure-based coarse-graining [124]. A GROMACS implementation can be found at code.google.com/p/plumx.