Chapter 7

Run parameters and Programs

7.1 On-line and HTML manuals

All the information in this chapter can also be found in HTML format in your GROMACS data directory. The path depends on where your files are installed, but the default location is `/usr/local/gromacs/share/gromacs/html/online.html`. If you installed from Linux packages it can typically be found as `/usr/share/gromacs/html/online.html`. You can also use the online manual from the GROMACS web site, `http://manual.gromacs.org/current`.

In addition, we install standard UNIX man pages for all the programs. If you have sourced the `GMXRC` script in the GROMACS binary directory for your host they should already be present in your `MANPATH` environment variable, and you should be able to type *e.g.* `man gmx-grompp`. You can also use the `-h` flag on the command line (e.g. `gmx grompp -h`) to see the same information, as well as `gmx help grompp`. The list of all programs are available from `gmx help`.

7.2 File types

Table 7.1 lists the file types used by GROMACS along with a short description, and you can find a more detail description for each file in your HTML reference, or in our online version.

GROMACS files written in XDR format can be read on any architecture with GROMACS version 1.6 or later if the configuration script found the XDR libraries on your system. They should always be present on UNIX since they are necessary for NFS support.

Default Name	Ext.	Type	Default Option	Description
atomtp	.atp	Asc		Atomtype file used by pdb2gmx
eiwit	.brk	Asc	-f	Brookhaven data bank file
state	.cpt	xdr		Checkpoint file
nnnice	.dat	Asc		Generic data file
user	.dlg	Asc		Dialog Box data for ngmx
sam	.edi	Asc		ED sampling input
sam	.edo	Asc		ED sampling output
ener	.edr			Generic energy: edr ene
ener	.edr	xdr		Energy file in portable xdr format
ener	.ene	Bin		Energy file
eiwit	.ent	Asc	-f	Entry in the protein date bank
plot	.eps	Asc		Encapsulated PostScript (tm) file
conf	.esp	Asc	-c	Coordinate file in ESPResSo format
conf	.g96	Asc	-c	Coordinate file in Gromos-96 format
conf	.gro	Asc	-c	Coordinate file in Gromos-87 format
conf	.gro		-c	Structure: gro g96 pdb esp tpr tpb tpa
out	.gro		-o	Structure: gro g96 pdb esp
polar	.hdb	Asc		Hydrogen data base
topinc	.itp	Asc		Include file for topology
run	.log	Asc	-l	Log file
ps	.m2p	Asc		Input file for mat2ps
ss	.map	Asc		File that maps matrix data to colors
ss	.mat	Asc		Matrix Data file
grompp	.mdp	Asc	-f	grompp input file with MD parameters
hessian	.mtx	Bin	-m	Hessian matrix
index	.ndx	Asc	-n	Index file
hello	.out	Asc	-o	Generic output file
eiwit	.pdb	Asc	-f	Protein data bank file
residue	.rtp	Asc		Residue Type file used by pdb2gmx
doc	.tex	Asc	-o	LaTeX file
topol	.top	Asc	-p	Topology file
topol	.tpb	Bin	-s	Binary run input file
topol	.tpr		-s	Generic run input: tpr tpb tpa
topol	.tpr		-s	Structure+mass(db): tpr tpb tpa gro g96 pdb
topol	.tpr	xdr	-s	Portable xdr run input file
traj	.trj	Bin		Trajectory file (architecture specific)
traj	.trr			Full precision trajectory: trr trj cpt
traj	.trr	xdr		Trajectory in portable xdr format
root	.xpm	Asc		X PixMap compatible matrix file
traj	.xtc		-f	Trajec., input: xtc trr trj cpt gro g96 pdb
traj	.xtc		-f	Trajectory, output: xtc trr trj gro g96 pdb
traj	.xtc	xdr		Compressed trajectory (portable xdr format)
graph	.xvg	Asc	-o	xvgr/xmgr file

Table 7.1: The GROMACS file types.

7.3 Run Parameters

7.3.1 General

Default values are given in parentheses. The first option in the list is always the default option. Units are given in square brackets The difference between a dash and an underscore is ignored. A sample `.mdp` file is available. This should be appropriate to start a normal simulation. Edit it to suit your specific needs and desires.

7.3.2 Preprocessing

`include:`
directories to include in your topology. Format:
`-I/home/john/mylib -I../otherlib`

`define:`
defines to pass to the preprocessor, default is no defines. You can use any defines to control options in your customized topology files. Options that are already available by default are:

`-DFLEXIBLE`
Will tell `grompp` to include flexible water in stead of rigid water into your topology, this can be useful for normal mode analysis.

`-DPOSRES`
Will tell `grompp` to include posre.itp into your topology, used for position restraints.

7.3.3 Run control

`integrator:` (Despite the name, this list includes algorithms that are not actually integrators. `steep` and all entries following it are in this category)

`md`
A leap-frog algorithm for integrating Newton's equations of motion.

`md-vv`
A velocity Verlet algorithm for integrating Newton's equations of motion. For constant NVE simulations started from corresponding points in the same trajectory, the trajectories are analytically, but not binary, identical to the `md` leap-frog integrator. The the kinetic energy, which is determined from the whole step velocities and is therefore slightly too high. The advantage of this integrator is more accurate, reversible Nose-Hoover and Parrinello-Rahman coupling integration based on Trotter expansion, as well as (slightly too small) full step velocity output. This all comes at the cost off extra computation, especially with constraints and extra communication in parallel. Note that for nearly all production simulations the `md` integrator is accurate enough.

md-vv-avek
A velocity Verlet algorithm identical to `md-vv`, except that the kinetic energy is determined as the average of the two half step kinetic energies as in the `md` integrator, and this thus more accurate. With Nose-Hoover and/or Parrinello-Rahman coupling this comes with a slight increase in computational cost.

sd
An accurate and efficient leap-frog stochastic dynamics integrator. With constraints, coordinates needs to be constrained twice per integration step. Depending on the computational cost of the force calculation, this can take a significant part of the simulation time. The temperature for one or more groups of atoms (`tc-grps`) is set with `ref-t` [K], the inverse friction constant for each group is set with `tau-t` [ps]. The parameter `tcoupl` is ignored. The random generator is initialized with `ld-seed`. When used as a thermostat, an appropriate value for `tau-t` is 2 ps, since this results in a friction that is lower than the internal friction of water, while it is high enough to remove excess heat NOTE: temperature deviations decay twice as fast as with a Berendsen thermostat with the same `tau-t`.

sd2
This used to be the default sd integrator, but is now deprecated. Four Gaussian random numbers are required per coordinate per step. With constraints, the temperature will be slightly too high.

bd
An Euler integrator for Brownian or position Langevin dynamics, the velocity is the force divided by a friction coefficient (`bd-fric` [amu ps^{-1}]) plus random thermal noise (`ref-t`). When `bd-fric=0`, the friction coefficient for each particle is calculated as mass/`tau-t`, as for the integrator `sd`. The random generator is initialized with `ld-seed`.

steep
A steepest descent algorithm for energy minimization. The maximum step size is `emstep` [nm], the tolerance is `emtol` [kJ mol^{-1} nm^{-1}].

cg
A conjugate gradient algorithm for energy minimization, the tolerance is `emtol` [kJ mol^{-1} nm^{-1}]. CG is more efficient when a steepest descent step is done every once in a while, this is determined by `nstcgsteep`. For a minimization prior to a normal mode analysis, which requires a very high accuracy, GROMACS should be compiled in double precision.

l-bfgs
A quasi-Newtonian algorithm for energy minimization according to the low-memory Broyden-Fletcher-Goldfarb-Shanno approach. In practice this seems to converge faster than Conjugate Gradients, but due to the correction steps necessary it is not (yet) parallelized.

nm
Normal mode analysis is performed on the structure in the `tpr` file. GROMACS should be compiled in double precision.

tpi

Test particle insertion. The last molecule in the topology is the test particle. A trajectory should be provided with the `-rerun` option of `mdrun`. This trajectory should not contain the molecule to be inserted. Insertions are performed `nsteps` times in each frame at random locations and with random orientiations of the molecule. When `nstlist` is larger than one, `nstlist` insertions are performed in a sphere with radius `rtpi` around a the same random location using the same neighborlist (and the same long-range energy when `rvdw` or `rcoulomb>rlist`, which is only allowed for single-atom molecules). Since neighborlist construction is expensive, one can perform several extra insertions with the same list almost for free. The random seed is set with `ld-seed`. The temperature for the Boltzmann weighting is set with `ref-t`, this should match the temperature of the simulation of the original trajectory. Dispersion correction is implemented correctly for tpi. All relevant quantities are written to the file specified with the `-tpi` option of `mdrun`. The distribution of insertion energies is written to the file specified with the `-tpid` option of `mdrun`. No trajectory or energy file is written. Parallel tpi gives identical results to single node tpi. For charged molecules, using PME with a fine grid is most accurate and also efficient, since the potential in the system only needs to be calculated once per frame.

tpic

Test particle insertion into a predefined cavity location. The procedure is the same as for `tpi`, except that one coordinate extra is read from the trajectory, which is used as the insertion location. The molecule to be inserted should be centered at 0,0,0. Gromacs does not do this for you, since for different situations a different way of centering might be optimal. Also `rtpi` sets the radius for the sphere around this location. Neighbor searching is done only once per frame, `nstlist` is not used. Parallel tpic gives identical results to single node tpic.

tinit: (0) [ps]

starting time for your run (only makes sense for integrators `md`, `sd` and `bd`)

dt: (0.001) [ps]

time step for integration (only makes sense for integrators `md`, `sd` and `bd`)

nsteps: (0)

maximum number of steps to integrate or minimize, -1 is no maximum

init-step: (0)

The starting step. The time at an step i in a run is calculated as: t = `tinit` + `dt`*(`init-step` + i). The free-energy lambda is calculated as: lambda = `init-lambda` + `delta-lambda`*(`init-step` + i). Also non-equilibrium MD parameters can depend on the step number. Thus for exact restarts or redoing part of a run it might be necessary to set `init-step` to the step number of the restart frame. `gmx convert-tpr` does this automatically.

comm-mode:

Linear

Remove center of mass translation

Angular
Remove center of mass translation and rotation around the center of mass

None
No restriction on the center of mass motion

nstcomm: (100) [steps]
frequency for center of mass motion removal

comm-grps:
group(s) for center of mass motion removal, default is the whole system

7.3.4 Langevin dynamics

bd-fric: (0) [amu ps^{-1}]
Brownian dynamics friction coefficient. When `bd-fric=0`, the friction coefficient for each particle is calculated as mass/`tau-t`.

ld-seed: (-1) [integer]
used to initialize random generator for thermal noise for stochastic and Brownian dynamics. When `ld-seed` is set to -1, a pseudo random seed is used. When running BD or SD on multiple processors, each processor uses a seed equal to `ld-seed` plus the processor number.

7.3.5 Energy minimization

emtol: (10.0) [kJ mol^{-1} nm^{-1}]
the minimization is converged when the maximum force is smaller than this value

emstep: (0.01) [nm]
initial step-size

nstcgsteep: (1000) [steps]
frequency of performing 1 steepest descent step while doing conjugate gradient energy minimization.

nbfgscorr: (10)
Number of correction steps to use for L-BFGS minimization. A higher number is (at least theoretically) more accurate, but slower.

7.3.6 Shell Molecular Dynamics

When shells or flexible constraints are present in the system the positions of the shells and the lengths of the flexible constraints are optimized at every time step until either the RMS force on the shells and constraints is less than emtol, or a maximum number of iterations (niter) has been reached

emtol: (10.0) [kJ mol^{-1} nm^{-1}]
the minimization is converged when the maximum force is smaller than this value. For shell MD this value should be 1.0 at most, but since the variable is used for energy minimization as well the default is 10.0.

niter: (20)
maximum number of iterations for optimizing the shell positions and the flexible constraints.

fcstep: (0) [ps^2]
the step size for optimizing the flexible constraints. Should be chosen as mu/(d^{2}V/dq^2) where mu is the reduced mass of two particles in a flexible constraint and d^{2}V/dq^2 is the second derivative of the potential in the constraint direction. Hopefully this number does not differ too much between the flexible constraints, as the number of iterations and thus the runtime is very sensitive to `fcstep`. Try several values!

7.3.7 Test particle insertion

rtpi: (0.05) [nm]
the test particle insertion radius see integrators `tpi` and `tpic`

7.3.8 Output control

nstxout: (0) [steps]
number of steps that elapse between writing coordinates to output trajectory file, the last coordinates are always written

nstvout: (0) [steps]
number of steps that elapse between writing velocities to output trajectory, the last velocities are always written

nstfout: (0) [steps]
number of steps that elapse between writing forces to output trajectory.

nstlog: (1000) [steps]
number of steps that elapse between writing energies to the log file, the last energies are always written

nstcalcenergy: (100)
number of steps that elapse between calculating the energies, 0 is never. This option is only relevant with dynamics. With a twin-range cut-off setup `nstcalcenergy` should be equal to or a multiple of `nstlist`. This option affects the performance in parallel simulations, because calculating energies requires global communication between all processes which can become a bottleneck at high parallelization.

nstenergy: (1000) [steps]
number of steps that else between writing energies to energy file, the last energies are always written, should be a multiple of `nstcalcenergy`. Note that the exact sums and

fluctuations over all MD steps modulo `nstcalcenergy` are stored in the energy file, so `g_energy` can report exact energy averages and fluctuations also when `nstenergy`>1

nstxout-compressed: (0) [steps]
number of steps that elapse between writing position coordinates using lossy compression

compressed-x-precision: (1000) [real]
precision with which to write to the compressed trajectory file

compressed-x-grps:
group(s) to write to the compressed trajectory file, by default the whole system is written (if `nstxout-compressed` > 0)

energygrps:
group(s) to write to energy file

7.3.9 Neighbor searching

cutoff-scheme:

Verlet
Generate a pair list with buffering. The buffer size is automatically set based on `verlet-buffer-tolerance`, unless this is set to -1, in which case `rlist` will be used. This option has an explicit, exact cut-off at `rvdw=rcoulomb`. Currently only cut-off, reaction-field, PME electrostatics and plain LJ are supported. Some `mdrun` functionality is not yet supported with the `Verlet` scheme, but `grompp` checks for this. Native GPU acceleration is only supported with `Verlet`. With GPU-accelerated PME or with separate PME ranks, `mdrun` will automatically tune the CPU/GPU load balance by scaling `rcoulomb` and the grid spacing. This can be turned off with `-notunepme`. `Verlet` is faster than `group` when there is no water, or if `group` would use a pair-list buffer to conserve energy.

group
Generate a pair list for groups of atoms. These groups correspond to the charge groups in the topology. This was the only cut-off treatment scheme before version 4.6. There is no explicit buffering of the pair list. This enables efficient force calculations for water, but energy is only conserved when a buffer is explicitly added.

nstlist: (10) [steps]

>0
Frequency to update the neighbor list (and the long-range forces, when using twin-range cut-offs). When this is 0, the neighbor list is made only once. With energy minimization the neighborlist will be updated for every energy evaluation when `nstlist`>0. With `cutoff-scheme=Verlet` and `verlet-buffer-tolerance` set, `nstlist` is actually a minimum value and `mdrun` might increase it, unless it is set to 1. With parallel simulations and/or non-bonded force calculation on the GPU, a value of 20

or 40 often gives the best performance. With `cutoff-scheme=Group` and non-exact cut-off's, `nstlist` will affect the accuracy of your simulation and it can not be chosen freely.

0

The neighbor list is only constructed once and never updated. This is mainly useful for vacuum simulations in which all particles see each other.

-1

Automated update frequency, only supported with `cutoff-scheme=group`. This can only be used with switched, shifted or user potentials where the cut-off can be smaller than `rlist`. One then has a buffer of size `rlist` minus the longest cut-off. The neighbor list is only updated when one or more particles have moved further than half the buffer size from the center of geometry of their charge group as determined at the previous neighbor search. Coordinate scaling due to pressure coupling or the `deform` option is taken into account. This option guarantees that their are no cut-off artifacts, but for larger systems this can come at a high computational cost, since the neighbor list update frequency will be determined by just one or two particles moving slightly beyond the half buffer length (which does not necessarily imply that the neighbor list is invalid), while 99.99% of the particles are fine.

nstcalclr: (-1) [steps]

Controls the period between calculations of long-range forces when using the group cut-off scheme.

1

Calculate the long-range forces every single step. This is useful to have separate neighbor lists with buffers for electrostatics and Van der Waals interactions, and in particular it makes it possible to have the Van der Waals cutoff longer than electrostatics (useful *e.g.* with PME). However, there is no point in having identical long-range cutoffs for both interaction forms and update them every step - then it will be slightly faster to put everything in the short-range list.

>1

Calculate the long-range forces every `nstcalclr` steps and use a multiple-time-step integrator to combine forces. This can now be done more frequently than `nstlist` since the lists are stored, and it might be a good idea *e.g.* for Van der Waals interactions that vary slower than electrostatics.

-1

Calculate long-range forces on steps where neighbor searching is performed. While this is the default value, you might want to consider updating the long-range forces more frequently.

Note that twin-range force evaluation might be enabled automatically by PP-PME load balancing. This is done in order to maintain the chosen Van der Waals interaction radius even if the load balancing is changing the electrostatics cutoff. If the `.mdp` file already specifies twin-range interactions (*e.g.* to evaluate Lennard-Jones interactions with a longer cutoff than the PME electrostatics every 2-3 steps), the load balancing will have also a small effect on

Lennard-Jones, since the short-range cutoff (inside which forces are evaluated every step) is changed.

ns-type:

grid
Make a grid in the box and only check atoms in neighboring grid cells when constructing a new neighbor list every `nstlist` steps. In large systems grid search is much faster than simple search.

simple
Check every atom in the box when constructing a new neighbor list every `nstlist` steps (only with `cutoff-scheme=group`).

pbc:

xyz
Use periodic boundary conditions in all directions.

no
Use no periodic boundary conditions, ignore the box. To simulate without cut-offs, set all cut-offs to 0 and `nstlist=0`. For best performance without cut-offs on a single MPI rank, use `nstlist=0`, `ns-type=simple`

xy
Use periodic boundary conditions in x and y directions only. This works only with `ns-type=grid` and can be used in combination with `walls`. Without walls or with only one wall the system size is infinite in the z direction. Therefore pressure coupling or Ewald summation methods can not be used. These disadvantages do not apply when two walls are used.

periodic-molecules:

no
molecules are finite, fast molecular PBC can be used

yes
for systems with molecules that couple to themselves through the periodic boundary conditions, this requires a slower PBC algorithm and molecules are not made whole in the output

verlet-buffer-tolerance: (0.005) [kJ/mol/ps]
Useful only with `cutoff-scheme=Verlet`. This sets the maximum allowed error for pair interactions per particle caused by the Verlet buffer, which indirectly sets `rlist`. As both `nstlist` and the Verlet buffer size are fixed (for performance reasons), particle pairs not in the pair list can occasionally get within the cut-off distance during `nstlist`-1 nsteps. This causes very small jumps in the energy. In a constant-temperature ensemble, these very small energy jumps can be estimated for a given cut-off and `rlist`. The estimate assumes a homogeneous particle distribution, hence the errors might be slightly underestimated for multi-phase systems. For longer pair-list life-time (`nstlist`-1)*dt the buffer is overestimated, because the interactions between particles are ignored. Combined with cancellation

of errors, the actual drift of the total energy is usually one to two orders of magnitude smaller. Note that the generated buffer size takes into account that the GROMACS pair-list setup leads to a reduction in the drift by a factor 10, compared to a simple particle-pair based list. Without dynamics (energy minimization etc.), the buffer is 5% of the cut-off. For NVE simulations the initial temperature is used, unless this is zero, in which case a buffer of 10% is used. For NVE simulations the tolerance usually needs to be lowered to achieve proper energy conservation on the nanosecond time scale. To override the automated buffer setting, use `verlet-buffer-tolerance`=-1 and set `rlist` manually.

`rlist: (1) [nm]`
Cut-off distance for the short-range neighbor list. With `cutoff-scheme=Verlet`, this is by default set by the `verlet-buffer-tolerance` option and the value of `rlist` is ignored.

`rlistlong: (-1) [nm]`
Cut-off distance for the long-range neighbor list. This parameter is only relevant for a twin-range cut-off setup with switched potentials. In that case a buffer region is required to account for the size of charge groups. In all other cases this parameter is automatically set to the longest cut-off distance.

7.3.10 Electrostatics

`coulombtype:`

`Cut-off`
Twin range cut-offs with neighborlist cut-off `rlist` and Coulomb cut-off `rcoulomb`, where `rcoulomb`≥`rlist`.

`Ewald`
Classical Ewald sum electrostatics. The real-space cut-off `rcoulomb` should be equal to `rlist`. Use *e.g.* `rlist=0.9`, `rcoulomb=0.9`. The highest magnitude of wave vectors used in reciprocal space is controlled by `fourierspacing`. The relative accuracy of direct/reciprocal space is controlled by `ewald-rtol`.
NOTE: Ewald scales as O($N^{3/2}$) and is thus extremely slow for large systems. It is included mainly for reference - in most cases PME will perform much better.

`PME`
Fast smooth Particle-Mesh Ewald (SPME) electrostatics. Direct space is similar to the Ewald sum, while the reciprocal part is performed with FFTs. Grid dimensions are controlled with `fourierspacing` and the interpolation order with `pme-order`. With a grid spacing of 0.1 nm and cubic interpolation the electrostatic forces have an accuracy of 2-3*10^{-4}. Since the error from the vdw-cutoff is larger than this you might try 0.15 nm. When running in parallel the interpolation parallelizes better than the FFT, so try decreasing grid dimensions while increasing interpolation.

`P3M-AD`
Particle-Particle Particle-Mesh algorithm with analytical derivative for for long range electrostatic interactions. The method and code is identical to SPME, except that the influence function is optimized for the grid. This gives a slight increase in accuracy.

Reaction-Field electrostatics
Reaction field with Coulomb cut-off `rcoulomb`, where `rcoulomb` $\geq$ `rlist`. The dielectric constant beyond the cut-off is `epsilon-rf`. The dielectric constant can be set to infinity by setting `epsilon-rf=0`.

Generalized-Reaction-Field
Generalized reaction field with Coulomb cut-off `rcoulomb`, where `rcoulomb` $\geq$ `rlist`. The dielectric constant beyond the cut-off is `epsilon-rf`. The ionic strength is computed from the number of charged (*i.e.* with non zero charge) charge groups. The temperature for the GRF potential is set with `ref-t` [K].

Reaction-Field-zero
In GROMACS, normal reaction-field electrostatics with `cutoff-scheme=group` leads to bad energy conservation. `Reaction-Field-zero` solves this by making the potential zero beyond the cut-off. It can only be used with an infinite dielectric constant (`epsilon-rf=0`), because only for that value the force vanishes at the cut-off. `rlist` should be 0.1 to 0.3 nm larger than `rcoulomb` to accommodate for the size of charge groups and diffusion between neighbor list updates. This, and the fact that table lookups are used instead of analytical functions make `Reaction-Field-zero` computationally more expensive than normal reaction-field.

Reaction-Field-nec
The same as `Reaction-Field`, but implemented as in GROMACS versions before 3.3. No reaction-field correction is applied to excluded atom pairs and self pairs. The 1-4 interactions are calculated using a reaction-field. The missing correction due to the excluded pairs that do not have a 1-4 interaction is up to a few percent of the total electrostatic energy and causes a minor difference in the forces and the pressure.

Shift
Analogous to `Shift` for `vdwtype`. You might want to use `Reaction-Field-zero` instead, which has a similar potential shape, but has a physical interpretation and has better energies due to the exclusion correction terms.

Encad-Shift
The Coulomb potential is decreased over the whole range, using the definition from the Encad simulation package.

Switch
Analogous to `Switch` for `vdwtype`. Switching the Coulomb potential can lead to serious artifacts, advice: use `Reaction-Field-zero` instead.

User
`mdrun` will now expect to find a file `table.xvg` with user-defined potential functions for repulsion, dispersion and Coulomb. When pair interactions are present, `mdrun` also expects to find a file `tablep.xvg` for the pair interactions. When the same interactions should be used for non-bonded and pair interactions the user can specify the same file name for both table files. These files should contain 7 columns: the `x` value, `f(x)`, `-f'(x)`, `g(x)`, `-g'(x)`, `h(x)`, `-h'(x)`, where `f(x)` is the Coulomb function, `g(x)` the dispersion function and `h(x)` the repulsion function. When `vdwtype` is not set to `User` the values for `g`, `-g'`, `h` and `-h'` are ignored. For the non-bonded interactions `x` values should run from 0 to the largest cut-off distance

+ `table-extension` and should be uniformly spaced. For the pair interactions the table length in the file will be used. The optimal spacing, which is used for non-user tables, is `0.002` [nm] when you run in mixed precision or `0.0005` [nm] when you run in double precision. The function value at `x=0` is not important. More information is in the printed manual.

`PME-Switch`
A combination of PME and a switch function for the direct-space part (see above). `rcoulomb` is allowed to be smaller than `rlist`. This is mainly useful constant energy simulations (note that using `PME` with `cutoff-scheme=Verlet` will be more efficient).

`PME-User`
A combination of PME and user tables (see above). `rcoulomb` is allowed to be smaller than `rlist`. The PME mesh contribution is subtracted from the user table by `mdrun`. Because of this subtraction the user tables should contain about 10 decimal places.

`PME-User-Switch`
A combination of PME-User and a switching function (see above). The switching function is applied to final particle-particle interaction, *i.e.* both to the user supplied function and the PME Mesh correction part.

`coulomb-modifier:`

`Potential-shift-Verlet`
Selects `Potential-shift` with the Verlet cutoff-scheme, as it is (nearly) free; selects `None` with the group cutoff-scheme.

`Potential-shift`
Shift the Coulomb potential by a constant such that it is zero at the cut-off. This makes the potential the integral of the force. Note that this does not affect the forces or the sampling.

`None`
Use an unmodified Coulomb potential. With the group scheme this means no exact cut-off is used, energies and forces are calculated for all pairs in the neighborlist.

`rcoulomb-switch: (0) [nm]`
where to start switching the Coulomb potential, only relevant when force or potential switching is used

`rcoulomb: (1) [nm]`
distance for the Coulomb cut-off

`epsilon-r: (1)`
The relative dielectric constant. A value of 0 means infinity.

`epsilon-rf: (0)`
The relative dielectric constant of the reaction field. This is only used with reaction-field electrostatics. A value of 0 means infinity.

7.3.11 VdW

vdwtype:

Cut-off
Twin range cut-offs with neighbor list cut-off `rlist` and VdW cut-off `rvdw`, where `rvdw` $\geq$ `rlist`.

PME
Fast smooth Particle-mesh Ewald (SPME) for VdW interactions. The grid dimensions are controlled with `fourierspacing` in the same way as for electrostatics, and the interpolation order is controlled with `pme-order`. The relative accuracy of direct/reciprocal space is controlled by `ewald-rtol-lj`, and the specific combination rules that are to be used by the reciprocal routine are set using `lj-pme-comb-rule`.

Shift
This functionality is deprecated and replaced by `vdw-modifier = Force-switch`. The LJ (not Buckingham) potential is decreased over the whole range and the forces decay smoothly to zero between `rvdw-switch` and `rvdw`. The neighbor search cut-off `rlist` should be 0.1 to 0.3 nm larger than `rvdw` to accommodate for the size of charge groups and diffusion between neighbor list updates.

Switch
This functionality is deprecated and replaced by `vdw-modifier = Potential-switch`. The LJ (not Buckingham) potential is normal out to `rvdw-switch`, after which it is switched off to reach zero at `rvdw`. Both the potential and force functions are continuously smooth, but be aware that all switch functions will give rise to a bulge (increase) in the force (since we are switching the potential). The neighbor search cut-off `rlist` should be 0.1 to 0.3 nm larger than `rvdw` to accommodate for the size of charge groups and diffusion between neighbor list updates.

Encad-Shift
The LJ (not Buckingham) potential is decreased over the whole range, using the definition from the Encad simulation package.

User
See `user` for `coulombtype`. The function value at `x=0` is not important. When you want to use LJ correction, make sure that `rvdw` corresponds to the cut-off in the user-defined function. When `coulombtype` is not set to `User` the values for `f` and `-f'` are ignored.

vdw-modifier:

Potential-shift-Verlet
Selects `Potential-shift` with the Verlet cutoff-scheme, as it is (nearly) free; selects `None` with the group cutoff-scheme.

Potential-shift
Shift the Van der Waals potential by a constant such that it is zero at the cut-off. This makes the potential the integral of the force. Note that this does not affect the forces or the sampling.

None
Use an unmodified Van der Waals potential. With the group scheme this means no exact cut-off is used, energies and forces are calculated for all pairs in the neighborlist.

Force-switch
Smoothly switches the forces to zero between `rvdw-switch` and `rvdw`. This shifts the potential shift over the whole range and switches it to zero at the cut-off. Note that this is more expensive to calculate than a plain cut-off and it is not required for energy conservation, since `Potential-shift` conserves energy just as well.

Potential-switch
Smoothly switches the potential to zero between `rvdw-switch` and `rvdw`. Note that this introduces articifically large forces in the switching region and is much more expensive to calculate. This option should only be used if the force field you are using requires this.

rvdw-switch: (0) [nm]
where to start switching the LJ force and possibly the potential, only relevant when force or potential switching is used

rvdw: (1) [nm]
distance for the LJ or Buckingham cut-off

DispCorr:

no
don't apply any correction

EnerPres
apply long range dispersion corrections for Energy and Pressure

Ener
apply long range dispersion corrections for Energy only

7.3.12 Tables

table-extension: (1) [nm]
Extension of the non-bonded potential lookup tables beyond the largest cut-off distance. The value should be large enough to account for charge group sizes and the diffusion between neighbor-list updates. Without user defined potential the same table length is used for the lookup tables for the 1-4 interactions, which are always tabulated irrespective of the use of tables for the non-bonded interactions. The value of `table-extension` in no way affects the values of `rlist`, `rcoulomb`, or `rvdw`.

energygrp-table:
When user tables are used for electrostatics and/or VdW, here one can give pairs of energy groups for which seperate user tables should be used. The two energy groups will be appended to the table file name, in order of their definition in `energygrps`, seperated by underscores. For example, if `energygrps = Na Cl Sol` and `energygrp-table = Na Na Na Cl`, `mdrun` will read `table_Na_Na.xvg` and `table_Na_Cl.xvg` in addition to the normal `table.xvg` which will be used for all other energy group pairs.

7.3.13 Ewald

fourierspacing: (0.12) [nm]
For ordinary Ewald, the ratio of the box dimensions and the spacing determines a lower bound for the number of wave vectors to use in each (signed) direction. For PME and P3M, that ratio determines a lower bound for the number of Fourier-space grid points that will be used along that axis. In all cases, the number for each direction can be overridden by entering a non-zero value for `fourier_n[xyz]`. For optimizing the relative load of the particle-particle interactions and the mesh part of PME, it is useful to know that the accuracy of the electrostatics remains nearly constant when the Coulomb cut-off and the PME grid spacing are scaled by the same factor.

fourier-nx (0) ; fourier-ny (0) ; fourier-nz: (0)
Highest magnitude of wave vectors in reciprocal space when using Ewald. Grid size when using PME or P3M. These values override `fourierspacing` per direction. The best choice is powers of 2, 3, 5 and 7. Avoid large primes.

pme-order (4)
Interpolation order for PME. 4 equals cubic interpolation. You might try 6/8/10 when running in parallel and simultaneously decrease grid dimension.

ewald-rtol (1e-5)
The relative strength of the Ewald-shifted direct potential at `rcoulomb` is given by `ewald-rtol`. Decreasing this will give a more accurate direct sum, but then you need more wave vectors for the reciprocal sum.

ewald-rtol-lj (1e-3)
When doing PME for VdW-interactions, `ewald-rtol-lj` is used to control the relative strength of the dispersion potential at `rvdw` in the same way as `ewald-rtol` controls the electrostatic potential.

lj-pme-comb-rule (Geometric)
The combination rules used to combine VdW-parameters in the reciprocal part of LJ-PME. Geometric rules are much faster than Lorentz-Berthelot and usually the recommended choice, even when the rest of the force field uses the Lorentz-Berthelot rules.

Geometric
Apply geometric combination rules

Lorentz-Berthelot
Apply Lorentz-Berthelot combination rules

ewald-geometry: (3d)

3d
The Ewald sum is performed in all three dimensions.

3dc
The reciprocal sum is still performed in 3D, but a force and potential correction applied in the `z` dimension to produce a pseudo-2D summation. If your system has a slab geometry in the `x-y` plane you can try to increase the `z`-dimension of the box (a box height of 3 times the slab height is usually ok) and use this option.

epsilon-surface: (0)
This controls the dipole correction to the Ewald summation in 3D. The default value of zero means it is turned off. Turn it on by setting it to the value of the relative permittivity of the imaginary surface around your infinite system. Be careful - you shouldn't use this if you have free mobile charges in your system. This value does not affect the slab 3DC variant of the long range corrections.

7.3.14 Temperature coupling

tcoupl:

no
No temperature coupling.

berendsen
Temperature coupling with a Berendsen-thermostat to a bath with temperature `ref-t` [K], with time constant `tau-t` [ps]. Several groups can be coupled separately, these are specified in the `tc-grps` field separated by spaces.

nose-hoover
Temperature coupling using a Nose-Hoover extended ensemble. The reference temperature and coupling groups are selected as above, but in this case `tau-t` [ps] controls the period of the temperature fluctuations at equilibrium, which is slightly different from a relaxation time. For NVT simulations the conserved energy quantity is written to energy and log file.

andersen
Temperature coupling by randomizing a fraction of the particles at each timestep. Reference temperature and coupling groups are selected as above. `tau-t` is the average time between randomization of each molecule. Inhibits particle dynamics somewhat, but little or no ergodicity issues. Currently only implemented with velocity Verlet, and not implemented with constraints.

andersen-massive
Temperature coupling by randomizing all particles at infrequent timesteps. Reference temperature and coupling groups are selected as above. `tau-t` is the time between randomization of all molecules. Inhibits particle dynamics somewhat, but little or no ergodicity issues. Currently only implemented with velocity Verlet.

v-rescale
Temperature coupling using velocity rescaling with a stochastic term (JCP 126, 014101). This thermostat is similar to Berendsen coupling, with the same scaling using `tau-t`, but the stochastic term ensures that a proper canonical ensemble is generated. The random seed is set with `ld-seed`. This thermostat works correctly even for `tau-t=0`. For NVT simulations the conserved energy quantity is written to the energy and log file.

nsttcouple: (-1)
The frequency for coupling the temperature. The default value of -1 sets `nsttcouple`

equal to nstlist, unless nstlist$\leq$0, then a value of 10 is used. For velocity Verlet integrators nsttcouple is set to 1.

nh-chain-length (10)
the number of chained Nose-Hoover thermostats for velocity Verlet integrators, the leap-frog md integrator only supports 1. Data for the NH chain variables is not printed to the .edr, but can be using the GMX_NOSEHOOVER_CHAINS environment variable

tc-grps:
groups to couple separately to temperature bath

tau-t: [ps]
time constant for coupling (one for each group in tc-grps), -1 means no temperature coupling

ref-t: [K]
reference temperature for coupling (one for each group in tc-grps)

7.3.15 Pressure coupling

pcoupl:

no
No pressure coupling. This means a fixed box size.

berendsen
Exponential relaxation pressure coupling with time constant tau-p [ps]. The box is scaled every timestep. It has been argued that this does not yield a correct thermodynamic ensemble, but it is the most efficient way to scale a box at the beginning of a run.

Parrinello-Rahman
Extended-ensemble pressure coupling where the box vectors are subject to an equation of motion. The equation of motion for the atoms is coupled to this. No instantaneous scaling takes place. As for Nose-Hoover temperature coupling the time constant tau-p [ps] is the period of pressure fluctuations at equilibrium. This is probably a better method when you want to apply pressure scaling during data collection, but beware that you can get very large oscillations if you are starting from a different pressure. For simulations where the exact fluctation of the NPT ensemble are important, or if the pressure coupling time is very short it may not be appropriate, as the previous time step pressure is used in some steps of the GROMACS implementation for the current time step pressure.

MTTK
Martyna-Tuckerman-Tobias-Klein implementation, only useable with md-vv or md-vv-avek, very similar to Parrinello-Rahman. As for Nose-Hoover temperature coupling the time constant tau-p [ps] is the period of pressure fluctuations at equilibrium. This is probably a better method when you want to apply pressure scaling during data collection, but beware that you can get very large oscillations if you are starting from a different pressure. Currently only supports isotropic scaling.

pcoupltype:

isotropic
Isotropic pressure coupling with time constant `tau-p` [ps]. The compressibility and reference pressure are set with `compressibility` [bar^{-1}] and `ref-p` [bar], one value is needed.

semiisotropic
Pressure coupling which is isotropic in the `x` and `y` direction, but different in the `z` direction. This can be useful for membrane simulations. 2 values are needed for `x/y` and `z` directions respectively.

anisotropic
Idem, but 6 values are needed for `xx`, `yy`, `zz`, `xy/yx`, `xz/zx` and `yz/zy` components, respectively. When the off-diagonal compressibilities are set to zero, a rectangular box will stay rectangular. Beware that anisotropic scaling can lead to extreme deformation of the simulation box.

surface-tension
Surface tension coupling for surfaces parallel to the xy-plane. Uses normal pressure coupling for the `z`-direction, while the surface tension is coupled to the `x/y` dimensions of the box. The first `ref-p` value is the reference surface tension times the number of surfaces [bar nm], the second value is the reference `z`-pressure [bar]. The two `compressibility` [bar^{-1}] values are the compressibility in the `x/y` and `z` direction respectively. The value for the `z`-compressibility should be reasonably accurate since it influences the convergence of the surface-tension, it can also be set to zero to have a box with constant height.

nstpcouple: (-1)
The frequency for coupling the pressure. The default value of -1 sets `nstpcouple` equal to `nstlist`, unless `nstlist` $\leq$0, then a value of 10 is used. For velocity Verlet integrators `nstpcouple` is set to 1.

tau-p: (1) [ps]
time constant for coupling

compressibility: [bar^{-1}]
compressibility (NOTE: this is now really in bar^{-1}) For water at 1 atm and 300 K the compressibility is 4.5e-5 [bar^{-1}].

ref-p: [bar]
reference pressure for coupling

refcoord-scaling:

no
The reference coordinates for position restraints are not modified. Note that with this option the virial and pressure will depend on the absolute positions of the reference coordinates.

all

The reference coordinates are scaled with the scaling matrix of the pressure coupling.

com

Scale the center of mass of the reference coordinates with the scaling matrix of the pressure coupling. The vectors of each reference coordinate to the center of mass are not scaled. Only one COM is used, even when there are multiple molecules with position restraints. For calculating the COM of the reference coordinates in the starting configuration, periodic boundary conditions are not taken into account.