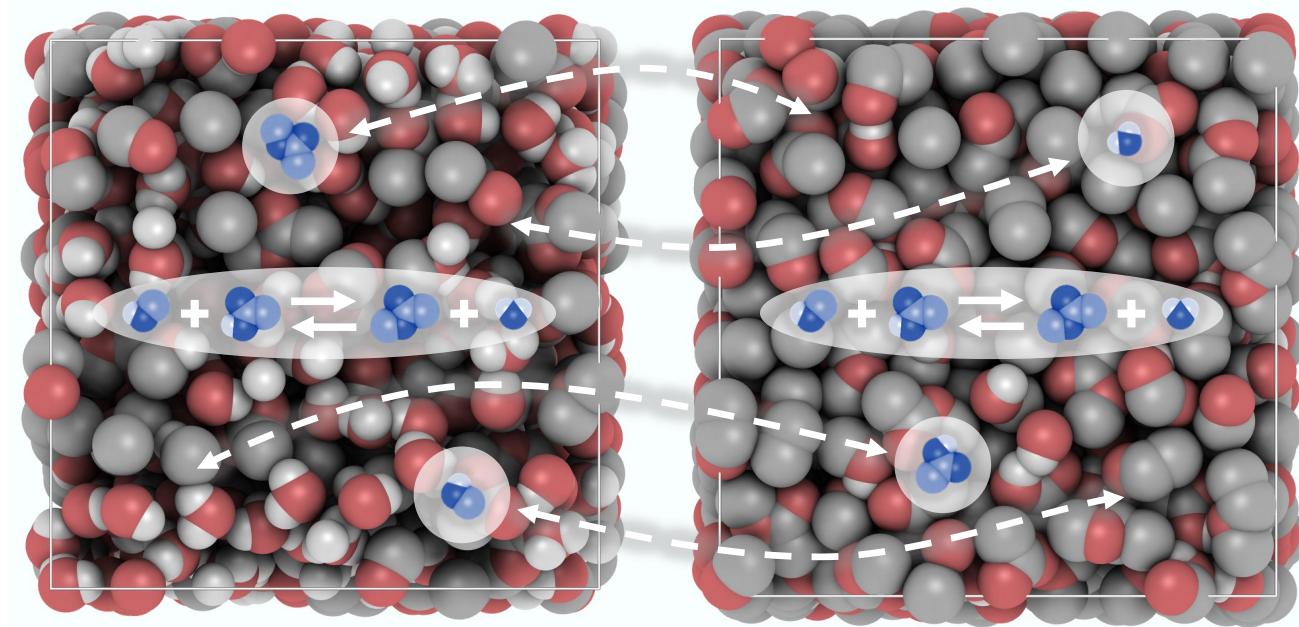


Brick-CFCMC

Open Source Software for Monte Carlo Simulations of
Phase and Reaction Equilibria



Remco Hens¹, Ahmadreza Rahbari¹, Sebastián Caro-Ortiz¹,
Noura Dawass¹, Máté Erdős¹, H. Mert Polat¹, Ali Poursaeidesfahani¹,
Hirad S. Salehi¹, Alper T. Celebi¹, Mahinder Ramdin¹,
Othon A. Moults¹, David Dubbeldam,² and Thijs J. H. Vlugt¹

¹Engineering Thermodynamics, Process and Energy Department, Faculty of Mechanical,
Maritime and Materials Engineering, Delft University of Technology

²Van 't Hoff Institute for Molecular Sciences, University of Amsterdam

Version 0.3.6

January 27, 2023

Copyright (c) 2021 R. Hens, A. Rahbari, S. Caro-Ortiz, N. Dawass,
M. Erdos, H.M. Polat, A. Poursaeidesfahani, H.S. Salehi, A.T. Celebi,
M. Ramdin, O.A. Moulton, D. Dubbeldam, and T.J.H. Vlugt

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Contents

1	First Things First	7
1.1	Overview	7
1.2	Obtaining Brick-CFCMC	7
1.3	Set up	8
1.4	Compiling	8
1.5	Updating	9
1.6	Running	9
1.7	Running an example	9
1.8	Creating your own simulation	10
2	INPUT	11
2.1	<code>settings.in</code>	11
2.1.1	Simulation Settings	11
2.1.2	Trial Move Probabilities	13
2.2	<code>forcefield.in</code>	14
2.2.1	Atom Types	14
2.2.2	Bending Types	14
2.2.3	Torsion Types	15
2.2.4	Energy Calculations	15
2.2.5	Lennard-Jones Interactions	16
2.2.6	Overrides	17
2.3	<code>topology.in</code>	17
2.3.1	Molecules	17
2.3.2	Partition Functions	17
2.3.3	Fugacity Coefficients	17
2.3.4	Fractional Groups	18
2.3.5	Reactions	18
2.3.6	Maximum Displacements	18
2.3.7	Pair and Cluster Settings	19
2.4	Molecule Definitions	19
2.5	<code>weightfunction.in</code>	20
2.6	<code>restart.in</code>	21
2.7	<code>wolfplot.in</code>	21
2.8	<code>umbrella.in</code>	21

3	OUTPUT	29
3.1	sim.log	29
3.2	System Properties	30
3.3	Averages	30
3.4	Configurations	30
3.5	Radial Distribution Functions	31
3.6	Umbrella Sampling	31
3.7	CFC	31
3.8	Weight function and Restart	32
3.9	CFCMC trial moves	33
3.10	Simulation Information	33
4	Commands	35
4.1	Overview	35
4.2	Detailed description	36
5	Energy	39
5.1	Intermolecular Lennard-Jones Interactions	39
5.1.1	Mixing Rules	39
5.1.2	Shifting the Potential	40
5.1.3	Analytic Tail Corrections	40
5.1.4	Scaling for Fractional Molecules	40
5.2	Intermolecular Electrostatic Interactions	41
5.2.1	Ewald Method	41
5.2.2	Wolf Method	42
5.2.3	Fennell-Gezelter (Damped Shifted Force Potential)	44
5.2.4	Scaling for Fractional Molecules	44
5.3	Intramolecular Lennard-Jones Interactions	45
5.4	Intramolecular Electrostatic Interactions	45
5.5	Bending	45
5.6	Torsion	45
5.7	Efficient Scaling of Fractional Interactions	46
6	Ensembles	51
6.1	Detailed balance	51
6.2	Scaled Coordinates	51
6.3	The <i>NVT</i> Ensemble	52
6.3.1	Partition Function	52
6.3.2	Acceptance Rules	52
6.4	The CFC <i>NVT</i> Ensemble	52
6.4.1	Partition Function	52
6.4.2	Acceptance Rules	53
6.4.3	Chemical Potential	53
6.5	The <i>NPT</i> Ensemble	54
6.5.1	Partition Function	54
6.5.2	Acceptance Rules	54
6.5.3	Chemical Potential	55

6.5.4	Partial Molar Enthalpy	55
6.5.5	Partial Molar Volume	55
6.6	The CFC <i>NPT</i> Ensemble	55
6.6.1	Partition Function	55
6.6.2	Acceptance Rules	56
6.6.3	Chemical Potential	57
6.6.4	Excess Chemical Potential from $\partial U/\partial \lambda$	57
6.6.5	Partial Molar Enthalpy	60
6.6.6	Partial Molar Volume	60
6.7	The CFC Gibbs Ensembles	60
6.7.1	Partition Function CFC Gibbs Ensemble at Constant Total Volume	61
6.7.2	Partition Function CFC Gibbs Ensemble at Constant Pressure	62
6.7.3	Acceptance Rules	62
6.7.4	Chemical Potential	65
6.8	The CFC Reaction Ensembles	65
6.8.1	Partition Function CFC Reaction Ensemble at Constant Volume	66
6.8.2	Partition Function CFC Reaction Ensemble at Constant Pressure	67
6.8.3	Acceptance Rules	67
6.8.4	Chemical Potential	71
6.8.5	Fugacity coefficient	71
6.9	The CFC Grand-Canonical and Osmotic Ensembles	72
6.9.1	Partition Function CFC Grand-Canonical Ensemble	72
6.9.2	Acceptance Rules	73
7	Trial Moves	75
7.1	Translation	75
7.1.1	Normal	75
7.1.2	Pair	75
7.1.3	Cluster	76
7.1.4	Smart	77
7.2	Rotation	78
7.2.1	Normal	78
7.2.2	Pair	79
7.2.3	Cluster	79
7.2.4	Smart	80
7.3	Volume Change	80
7.3.1	Normal	80
7.3.2	Cluster	82
7.4	Intramolecular	85
7.4.1	Bending	85
7.4.2	Torsion Move	85
7.5	Lambda Moves	86
7.5.1	Normal	86
7.5.2	Grand-Canonical Ensemble	86
7.6	CFC Swap (Reinsertion)	87
7.6.1	<i>NVT/NPT</i>	87

7.6.2	Gibbs Ensemble	88
7.6.3	Reaction Ensemble	88
7.7	CFC Identity Change	89
7.7.1	<i>NVT/NPT</i>	89
7.7.2	Gibbs Ensemble	89
7.7.3	Reaction Ensemble	90
7.8	CFC Hybrid Trial Moves	90
8	Examples	103
8.1	<i>NPT</i> Ensemble	103
8.2	Gibbs Ensemble	104
8.3	Reaction Ensemble	105
8.4	Grand-Canonical and Osmotic Ensemble	105
8.5	Ionic Liquids	106
9	Troubleshooting	109
9.1	Warning Messages	109
9.2	Error Messages	110
10	References	115

Chapter 1

First Things First

1.1 Overview

This code, Brick-CFCMC¹, is designed for performing Molecular Simulations of gases, liquids and their mixtures using state-of-the-art Continuous Fractional Component Monte Carlo techniques. In Brick-CFCMC, various ensembles can be combined: *NVT*, *NPT*, the Gibbs Ensemble, the Reaction Ensemble, the Grand Canonical Ensemble and the Osmotic Ensemble. Properties such as chemical potentials, fugacity coefficients, partial molar enthalpies and partial molar volumes can be directly obtained from single simulations.

Brick-CFCMC has been used in some of our simulation work such as the study of the ammonia synthesis reaction [1], the computation of partial molar enthalpies and reaction enthalpies [2], the computation of solubility of water in high-pressure hydrogen [3], the study of vapor-liquid equilibria of xylene mixtures [4, 5], the computation of thermodynamic properties of water-methanol mixtures, and many more.

In addition to this manual, we refer the reader to the two main publications where the functionality of Brick-CFCMC is described [6, 7], a recent review paper on the CFCMC technique [8], and several PhD theses of the Engineering Thermodynamics group in Delft [9, 10].

If you use Brick-CFCMC in your publication, please cite the following papers:

- J. Chem. Inf. Model., 2020, 60, 2678-2682 (Ref. [6])
- J. Chem. Inf. Model., 2021, 61, 3752-3757 (Ref. [7])
- Molecular Simulation, 2021, 47, 804-823 (Ref. [8])

1.2 Obtaining Brick-CFCMC

The Git repository containing the source code, molecules, force field parameters, tools, and more are located on Gitlab. If you want to obtain Brick-CFCMC you simply clone the repository:

```
git clone https://gitlab.com/ETH_TU_Delft/Brick-CFCMC.git ${HOME}/brick
```

¹Named after the character Brick Heck, from the TV series *The Middle*, whose parents gave him this name in the hope that a special name would make him do special things.

or, if you have set up a GitLab account:

```
git clone git@gitlab.com:ETH_TU_Delft/Brick-CFCMC.git ${HOME}/brick
```

which will create a directory `brick` in your home directory.

1.3 Set up

To use the Brick-CFCMC software properly the first time one has to do add following three lines to the file `.bashrc`: (we assume that bash is your default shell, and that you are using a modern Unix/Linux system)

```
export BRICK_DIR=${HOME}/brick
. ${BRICK_DIR}/.brick.sh
. ${BRICK_DIR}/.autocompletion
```

This implies the following:

- The location of the main directory of Brick-CFCMC is set in the variable `BRICK_DIR`. This is necessary for all commands to work: they need to know where the source files and executables are.
- Bash functions are added to the bash shell by sourcing the (hidden) file `.brick.sh` in your `.bashrc`. This means you can use commands (Chapter 4) that come with Brick-CFCMC.
- Autocompletion for the commands is used by sourcing the (hidden) file `.autocompletion` in your `.bashrc`.

Do not forget to reload your terminal for the changes to be updated. Either close and open a new terminal or execute:

```
source ~/.bashrc
```

1.4 Compiling

Brick-CFCMC can be compiled with the Intel Fortran Compiler [11] and the GNU Fortran Compiler [12]. We strongly recommend the use of the Intel Fortran Compiler. To compile the code for the first time one can simply run the command:

```
brick compile
```

This will compile the main code as well as the tools and create all executables with both the Intel Fortran Compiler and the GNU Fortran Compiler. If you do not have the Intel Fortran Compiler, please type the following instead:

```
brick compile -g
```

The executables are placed in the directory `${BRICK_DIR}/EXECUTABLES`. It is also possible to manually choose what you want to compile and how. For example, if you want to compile the main code using the Intel Fortran Compiler with all optimization you execute the command:

```
brick compile --ifort --optimize
```

For more details about the compiling options see Chapter 4 or run the command:

```
brick compile --help
```

If the Intel Fortran Compiler is present, then the tools will always be compiled using that compiler.

1.5 Updating

Updating Brick-CFCMC is done via the Git repository and can be achieved by the command:

```
brick update
```

Do not forget to compile again to update the executables.

1.6 Running

Running Brick-CFCMC is done by executing the `run` file that can be copied from the `RUN` directory. Place the `run` file in a directory together with an `INPUT` folder (Chapter 2) and execute:

```
./run
```

on the command line. By default this will run the optimized version compiled with the Intel Fortran Compiler in the background. However there are more options available:

options	action
<code>-i/--intel</code> / <code>--ifort</code>	run with the Intel Fortran compiler
<code>-g/--gnufortran</code> / <code>--gfortran</code>	run with the GNU Fortran compiler
<code>-o/--optimize</code>	run with optimizations (Intel or GNU)
<code>-d/--debug</code>	run with debug options (Intel or GNU)
<code>-t/--terminal</code>	print to the terminal instead of <code>sim.log</code>

By default the options `--intel` and `--optimize` are used.

1.7 Running an example

If you are a new user to Brick-CFCMC, we advise you to run one of the examples, provided in the directory `${BRICK_DIR}/EXAMPLES`. These examples are documented in detail in Chapter 8. For example, to compute the vapor-liquid equilibrium of methanol using the Gibbs Ensemble technique, please type

```
cd ${BRICK_DIR}/EXAMPLES/Gibbs_Ensemble/Example_2_VLE_Methanol
```

This directory will show a directory `INPUT` and an executable `run`. To start this simulation, please type

```
./run --terminal
```

which will run Brick-CFCMC and write output to the terminal. Please have a careful look at the input and output files.

1.8 Creating your own simulation

To set up your first simulation run yourself, please type first:

```
brick new MyFirstSimulation
```

and next, to create input files for the simulation:

```
brick input
```

This will execute the tool that helps you create the input files. When you have created the input files for your simulation you can run your simulation by typing:

```
./run --terminal
```

which will run Brick-CFCMC and write output to the terminal.

Chapter 2

INPUT

The INPUT folder contains all the files with input parameters for a simulation. For every simulation at least the following three files should be present: `settings.in`, `forcefield.in`, and `topology.in`. It is possible to create the files from scratch yourself, however it is advised to use the input creator tool or copy and modify files from the EXAMPLES. The input creator tool can be called from the command line by: `brick input`

2.1 `settings.in`

The `settings.in` file contains parameters to set up a simulation. The format is fixed and can only be read in this form (File 1). Here, we explain the meaning of all the parameters in this file.

2.1.1 Simulation Settings

- **Nbox** sets the number of simulation boxes.
- **Temperature** is the temperature of the system in K.
- **Pressure** is the applied pressure to the system. Different units can be given (Pa, kPa, hPa, MPa, bar, and red (for reduced units)) separated from the value by a whitespace. In case a negative number is supplied, this means that the (total) volume of the system is constant. This is required in the *NVT* version of the Gibbs ensemble. Setting the pressure to (exactly) zero can be useful for obtaining a quick guess of the coexisting liquid density (*NPT* ensemble) [13].
- **Reduced Units?**
 - .true. use reduced units: length in units of σ , energies in units of ϵ , temperature in units of ϵ/k_B , pressure in units of ϵ/σ^3 etc.
 - .false. use conventional units: length in units of Å, energy/ k_B in units of K, temperature in K, pressure in Pa, etc.
- **Nproduction** sets the number of production cycles in the simulation.
- **Nequilbrate** sets the number of equilibration cycles.

- **Ninitialize** sets the number of cycles to initialize the system.

When Nproduction, Nequilbrate or Ninitialize are set to 0 or less, no cycles in this stage of the simulation are performed. One cycle in the simulation consists of N MC steps, where N is the total number of molecules at the start of the simulation (with a minimum of 20). In the initialization stage only translation and rotation trial moves are used. This way, most of the overlaps between molecules are removed. Output is not written during this stage. During the initializing and equilibration stage, maximum displacement parameters (for the molecule translation, molecule rotation, and volume change trial moves) are modified to achieve an acceptance ratio of around 50 %.

The third row sets how often system properties (density, volume, etc.) should be updated and written to the output:

- **Nconfiguration** is the number of cycles after which the current configuration is written to the output (only during the initializing and production cycles)
- **Ndata** is the number of cycles after which averages are written to a file (only during the production cycles)
- **Naverage** is the number of cycles after which averages are updated in the simulation (only during the production cycles)
- **Nrdf** is the number of cycles after which the RDFs are updated in the simulation (only during the production cycles)

When Nconfiguration, Ndata, Naverage or Nrdf are set to 0 or less the updating is skipped. For example, if Nconfiguration is set to 0, no configuration files are written (except for the initial and final configuration).

- **Init?**
 - .true. a random initial configuration is created. A hard sphere cutoff is used to prevent overlaps between molecules.
 - .false. the initial configuration is read from the file `restart.in` (see also Sec. 2.6 and 3.8).
- **Weight?**
 - .true. a weight function is read from the file `weightfunction.in` (see also Sec. 2.5 and 3.8).
 - .false. do not use a weight function from a file.
- **Seed?**
 - .true. start the simulation with a fixed **Seed** for the Mersenne twister RNG [14].
 - .false. start the simulation with a random seed for the RNG.
- **RDF_Molecule?**
 - .true. calculate the RDF for molecules (using the geometric center of the molecules).
 - .false. do not calculate the RDF of molecules.
- **RDF_Atom?**
 - .true. calculate the RDF for atoms.
 - .false. do not calculate the RDF for atoms.

- **Insertions?**
 - .true. use Widom's test particle insertion method for calculation the chemical potential [15] and enthalpy [16].
 - .false. do not use insertion methods.
- **WolfPlot?**
 - .true. calculate energies for different parameters to create a 'Wolf plot' (see Sec. 5.2.2).
 - .false. run the program normally.

The following parameters are settings to use the Wang Landau scheme [17] to obtain a weight function for biasing λ to obtain a flat probability distribution. This only works in simulations that use the CFC method.

- **WL?**
 - .true. use the Wang Landau scheme [17] to construct a weight function.
 - .false. do not use the Wang Landau scheme (no weight function is constructed). Choose this option if you want to use a weight function from a file and you do not want to change it.
- **Fmod** is the modification factor in the Wang Landau scheme. The default value is 0.1.
- **Fred** is the reduction factor in the Wang Landau scheme. The default value is 4.0.
- **Flatc** is the flatness criterion in the Wang Landau scheme. The default value is 0.2.
- **Linear?** use a linear interpolation of the weightfunction? The default is .false.
- **Spline?** use a spline for interpolating the weightfunction? The default is .false.
- **dU/dL?** calculate $\langle \frac{\partial U}{\partial X} \rangle$? The default is .false. **NOTE:** only use this option for simulations with exactly one fractional in the *NPT* ensemble. See also Sec. 6.6.4.

2.1.2 Trial Move Probabilities

The second part of the `input` file sets the probabilities to select a Monte Carlo trial move. The probabilities of the trial moves are automatically normalized. See Chapter 7 for more information on the trial moves.

- **Translation Moves** trial moves that attempt to translate a molecule, pair or cluster.
- **Rotation Moves** trial moves that attempt to rotate a molecule, pair or cluster.
- **Volume Moves** trial moves that attempt to change the volume of the simulation boxes. In the Gibbs ensemble, one can either change the volume of each simulation box independently (*NPT* version of the Gibbs ensemble, only for systems with more than one molecule type), or change the volume of each simulation box so that the total volume is fixed (required for single components, and optional for other systems)
- **Intramolecular Moves** trial moves that attempt to change the geometry of a molecule.
- **Lambda Moves** trial moves that attempt to change the fractional parameter λ of a Fractional Group.

- **CFC Hybrid Moves** trial moves that combine the Swap and Identity Change moves.
- **Hybrid Move Switches** define the switching parameters for the Hybrid CFC moves. It contains six values: two for each ensemble. In the example in File 1, for $\lambda < 0.3$ a Swap trial move in the Gibbs Ensemble is performed, for $\lambda > 0.7$ an Identity Change trial move in the Gibbs Ensemble is performed, and otherwise no trial move is performed.

2.2 forcefield.in

The `forcefield.in` file contains the interaction parameters for the molecules in the simulation (File 2).

2.2.1 Atom Types

In the first block all atom types in the simulation are defined in the following form:

- **Atom** is a unique label of an atom type.
- **Sigma** is the Lennard-Jones interaction parameter σ in Å (or reduced units).
- **Epsilon** is the Lennard-Jones interaction parameter ϵ/k_B in K (or reduced units).
- **Charge** is the charge of the atom type in e .
- **LJ?** is a Boolean (T/F) that states if the atom type has Lennard-Jones interactions.
- **EL?** is a Boolean (T/F) that states if the atom type has Electrostatic interactions.
- **Print** is the character (element) that should be printed in a configuration file.

2.2.2 Bending Types

In the second block all bending types in the molecules are defined. The following interaction potential is used for the bending energy:

$$U_{\text{Bending}} = \frac{K}{2} (\theta - \theta_0)^2 \quad (2.1)$$

- **Bending** is a unique label of a bending type.
- **theta0** is the equilibrium angle θ_0 in degrees.
- **K** is the bending constant K in K/rad²
- **dtheta** is the maximum angle displacement for the bending trial move in degrees.

2.2.3 Torsion Types

In the third block all torsion types in the molecules are defined. There are three types of torsion potentials that can be used:

- TraPPE [18]

$$U_{\text{torsion}} = c_0 + c_1(1 + \cos \phi) + c_2(1 - \cos 2\phi) + c_3(1 + \cos 3\phi) \quad (2.2)$$

- OPLS [19]

$$U_{\text{torsion}} = \frac{K_1}{2}(1 + \cos \phi) + \frac{K_2}{2}(1 - \cos 2\phi) + \frac{K_3}{2}(1 + \cos 3\phi) + \frac{K_4}{2}(1 - \cos 4\phi) \quad (2.3)$$

- Ryckaert-Bellemans [20]

$$U_{\text{torsion}} = \sum_{i=0}^5 p_i \cos^i(\phi) \quad (2.4)$$

- **Torsion** is a unique label of a torsion type. If the first letter is a 'T' the TraPPE form is used, if it is a 'O' the OPLS form is used, and if it is a 'R' the Ryckaert-Bellemans form is used.
- **A0,...,A5** are the potential parameters ($c_0, \dots, c_3, K_1, \dots, K_4$ or p_0, \dots, p_5) in K. (Note that **no values** for A4 and A5 should be given for the TraPPE and OPLS forms because then A4 will be interpreted as the maximum torsion angle displacement.)
- **dphi** is the maximum torsion angle displacement for the torsion trial move in degrees.

2.2.4 Energy Calculations

After defining the atom, bending, and torsion types, parameters and settings for calculating energies and forces are defined. The following settings and parameters can be given (for each simulation box separately) in the fourth block:

- **Ideal_Gas** Consider an ideal gas in the simulation box? (.true. or .false.). The default value is .false..
- **Cutoff_LJ_Energy** The cutoff radius for the energy from Lennard-Jones interactions (in Å or reduced units). The default value is 14 Å.
- **Method_EL_Energy** The method for calculating the energy from electrostatic interactions:
 - Wolf the Wolf summation method [21].
 - FG the modified Wolf method by Fennell and Gezelter [22].
 - Ewald the Ewald method [23].
 - none no method. (A warning will be printed.)

The default is **none**. Note that **none** should also be used if one wants to perform a simulation of one single charged molecule (without any self interaction) because the Ewald and Wolf methods implicitly assume periodic images of the molecules.

- **Cutoff_EL_Energy** The cutoff radius for energy from electrostatic interactions in Å (or reduced units). The default is 14 Å.
- **Cutoff_EL_Force** The cutoff radius for the electrostatic forces (using the DSF potential [22]) in Å (or reduced units). Required for the smart trial moves. The default is the same value as **Cutoff_EL_Energy**. Note that this means that for the electrostatic force the same cut off is taken as for the electrostatic energy, unless explicitly defined differently. Also, in Brick-CFCMC the length of the simulation box might become smaller than twice the cut off for the electrostatic force (without giving an error), and one should make sure that this does not happen (for energy calculations an error is always given).
- **Alpha_EL_Energy** The damping parameter for energy calculations in the chosen electrostatic method in Å⁻¹. See Sections 5.2.1 and 5.2.2 for more details on how to chose the correct values. The default is 0.12 Å⁻¹.
- **Alpha_EL_Force** The damping parameter for electrostatic force calculations (using the DSF potential [22]) in Å⁻¹. Required for the smart trial moves. The default is 0.12 Å⁻¹.
- **Kmax_Ewald** The number of vectors in each direction for the Ewald method. The default is 10.
- **dt_SmartTranslation** The time step for the smart translation trial move (Sec. 7.1.4) in ps. The default is 0.01.
- **Nstep_SmartTranslation** The number of time steps used in the smart translation trial move (Sec. 7.1.4). The default is 10.
- **dt_SmartRotation** The time step for the smart rotation trial move (Sec. 7.2.4) in ps. The default is 0.01.
- **Nstep_SmartRotation** The number of time steps used in the smart rotation trial move (Sec. 7.2.4). The default is 10.

The (boldface) keywords can be put in any order.

2.2.5 Lennard-Jones Interactions

In the fifth block two additional settings can be defined.

- **LJ_Truncation**

Shifted	Truncate and shift the Lennard-Jones potential.
Tailcorrections	Truncate and apply analytic tail corrections.
none	Truncate without any correction. (A warning will be printed.)

The default is **Tailcorrections**.
- **Mixing_Rules**

Lorentz-Berthelot	Lorentz-Berthelot mixing rules for the Lennard-Jones interactions.
Jorgensen	Jorgensen mixing rules for the Lennard-Jones interactions.

The default is **Lorentz-Berthelot**.

2.2.6 Overrides

In the final block, overrides can be defined for Lennard-Jones interactions.

- **atom 1, atom 2** are the atom type numbers (i, j).
- **Sigma** is the new σ_{ij} (instead of using the mixing rules).
- **Epsilon** is the new ϵ_{ij} (instead of using the mixing rules).
- **Rmin** is $r_{\min,ij}$, the minimum distance between two atoms (see Section 5.1.1).

Note that if the interaction between atoms i and j is defined in the overrides, then the interaction parameters between j and i are automatically set to the same values.

2.3 topology.in

The `topology.in` sets the number of molecules in the system, defines Fractional Types, and sets the (initial) number of molecules and other system parameters. For an example see File 3.

2.3.1 Molecules

The first block sets the initial number of molecules in the simulation.

- **Molecule** is the name of the molecule type. It should match exactly with the name of a molecule definition file (see Sec. 2.4) in the INPUT folder. After the name, the number of molecules of this type in each simulation box is given.

2.3.2 Partition Functions

This block is optional and is used for simulations in the Reaction Ensemble.

- **Partition Functions** The first entry is the molecule name and the second entry is the natural logarithm of the partition function divided by the thermal wavelength cubed:

$$\ln \left[\frac{qV_0}{\Lambda^3} \right] \quad (2.5)$$

where $V_0 = 1 \text{ \AA}^3$ and the $\Lambda = \frac{\hbar}{\sqrt{2\pi Mk_B T}}$ with M the mass of one molecule. The default value of the expression in Eq. 2.5 is 0. The reader is referred to Ref. [10] on how to obtain values for isolated molecule partition functions for molecules.

2.3.3 Fugacity Coefficients

This block is optional and is used for simulations in the Grand Canonical Ensemble.

- **Fugacity Coefficients** The first entry is the molecule name and the second entry is the fugacity coefficient. The default value is 1.

2.3.4 Fractional Groups

In the next block Fractional Groups are defined. On the first line, the number of Fractional Groups is given (this excludes the fractionals used for Reactions).

- **Type** is the Fractional Type. It can be *NVT/NPT*, GE or GCMC.
- **Box** is the simulation box in which the Fractional Group is initially placed. Note that only for Fractionals of the GE type the box can change during the simulation.
- **NInFrac** is the number of molecules in the Fractional Group.
- **MoleculeTypes** is a list with the molecule names that are in the Fractional Group. If a Molecule Type is more than once in a Fractional Group the molecule name is repeated.
- **Name** is the name of the Fractional Group and is used to print in the output.

2.3.5 Reactions

After the block defining the Fractional Groups, reactions can be defined. It is advised to use the input creator tool for this. The first line sets the number of reactions, after which the reactions are defined in the following form:

```
N MolTypes in Reaction in Box M
MolType      Nstoi
MoleculeName_1    v_1
...
...
...
MoleculeName_N    v_N
```

For example the reaction $A + 4B \rightleftharpoons 5C \rightleftharpoons 2D + 3E (\rightleftharpoons A + 4B)$ is defined as:

```
5 MolTypes in Reaction in Box 1
MolType      Nstoi
A            -1
B            -4
C            5
D            -2
E            -3
```

This will result in the creation of three Fractional Types: $A + 4B$, $5C$ and $2D + 3E$ (one fractional per reaction step).

2.3.6 Maximum Displacements

In the next block, the boxsize(s) and maximum displacements for the trial moves are set.

- **Box** is the number of the box (starting from 1).
- **Length** is the length of the edge of the cubic simulation box in Å (or reduced units).

- **dVolume** is the maximum volume change for the volume change trial moves in Å³ (or reduced units).

Then, for each molecule type the maximum displacements for translation and rotation are set in each box.

- **MolType** is the name of the molecule type.
- **dtrans** is the maximum displacement for the translation trial move in Å (or reduced units).
- **drotation** is the maximum rotation angle for the rotation trial move in degrees.

And finally, for each Fractional Group, the maximum displacement in the fractional parameter λ is set.

- **Fractional** is the number of the Fractional Group (in the order of definition of the Fractional Groups).
- **dlambda** is the maximum change in the fractional parameter λ .

2.3.7 Pair and Cluster Settings

In the final block, molecule type pairs can be defined. First the number of pairs is set and then the names of the molecule types that form a pair are given. The last line defines the radius of a cluster in case cluster trial moves are used.

Fractional Molecules, Fractional Groups, Fractional Types and Molecule Pairs

A Fractional Group consists of Fractional Molecules. For example, in GE simulations of Ionic Liquids, a Fractional Group should be consisting of the anion and cation. This way molecule transfer between the two boxes always keeps boxes charge neutral because the fractional anion and fractional cation will be transferred simultaneously. Each Fractional Group is of a certain Fractional Type. Currently, there are four: *NVT/NPT*, GE, Reaction and GCMC. Molecule pairs consists out of Molecule Types. Those pair definitions are used in the pair trial moves; they have no relation with the Fractional Types. For example, in GE simulations of Ionic Liquids, a molecule pair could be the anion and cation. This way, pair translation moves will translate an anion and cation simultaneously.

2.4 Molecule Definitions

Each molecule type that is used during the simulation is defined in its own file. These files should be put in the INPUT folder. A template file **molecule.example** of a molecule definition can be found in the TEMPLATES folder (see also File 5). The first block states the molar mass of the molecule (in g/mol) and this value is used to convert densities in the output to kg/m³.

The second block defines the geometry of the molecule. The first entry is the atom label which should correspond to one of the labels in the **forcefield.in** file. The third, fourth and fifth entries are the *x*, *y* and *z* coordinates of the atom. The coordinates should always be given, even when the molecule is flexible. The third block defines which atoms share a bond, this is done via the

numbers of the atoms in the previous block. Defining bonds is important when bendings and/or torsions are used.

The third block defines the bendings in a molecule. The first entry is the label of the bending that should correspond to one of the labels in the `forcefield.in` file. The second, third and fourth entry specify the atom numbers that define a bending.

The fourth block defines the torsions in a molecule. The first entry is the label of the torsion that should correspond to one of the labels in the `forcefield.in` file. The second till the fifth entry specify the atom numbers that define a torsion.

The last block defines intramolecular interactions. The first two entries specify the atom numbers (i and j) that have an intramolecular interaction. The third, fourth and fifth entry are the scaling parameters α_{ij} , β_{ij} and γ_{ij} for the intramolecular interactions. The scaling parameters are used as follows (see also Sec. 5.3 and 5.4).

- Intramolecular Lennard-Jones interactions

$$U_{\text{LJ}} = 4\epsilon_{ij} \left[\alpha_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \beta_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (2.6)$$

- Intramolecular Electrostatic interactions

$$U_{\text{EL}} = \frac{1}{4\pi\epsilon_0} \frac{\gamma_{ij}q_iq_j}{r_{ij}} \quad (2.7)$$

Note that for atoms that do not have a intermolecular LJ interaction but do have an intramolecular LJ interaction the values $\sigma = 0.1$ and $\epsilon = 1\text{K}$ are taken for the intramolecular interaction. In this way, it is possible to define extra intramolecular interactions. For example, consider the repulsive term for alcohols in the TraPPE force field [24]:

$$u_{\text{repulsive}}(r_{ij}) = \frac{a_{\text{alcohol}}}{r_{ij}^{12}} \quad (2.8)$$

between the hydrogen atom in the hydroxyl group and the oxygen atom. In this force field the hydrogen atom does not have a LJ interaction and therefore $\epsilon_H = 1$ and $\sigma_H = 0.1$. We then have (with Lorentz-Berthelot mixing rules):

$$\alpha_{ij} = \frac{a_{\text{alcohol}}}{4\epsilon_{ij}\sigma_{ij}^{12}} = \frac{a_{\text{alcohol}}}{4\sqrt{\epsilon_o\epsilon_H} \left(\frac{1}{2}[\sigma_O + \sigma_H] \right)^{12}} \quad (2.9)$$

$$\beta_{ij} = 0 \quad (2.10)$$

2.5 weightfunction.in

This file can be added to the INPUT folder if one would like to use a weight function from a file. The file contains `Nlambdabins` blocks of rows with either one or two columns. The first column is the bin number and the second (and third) column(s) is(are) the weights. Each block defines the weight function for a different Fractional Group (the order is set by the order of definition in the `topology.in` file). Weight functions can be obtained from previous simulations (`weightfunction.out` in the OUTPUT folder). Remember to set `Lweight` to `.true.` if this file should be used. Moreover, in case that the weight function should not be modified by the Wang Landau algorithm, set `LWL` to `.false..`

2.6 restart.in

This file can be added to the INPUT folder if one would like a simulation to start from a specific configuration. Restart files can be obtained from previous simulations (`restart.out` in the OUTPUT folder). It contains the positions of all atoms in the simulation, boxesizes, maximum displacement values and fractional parameters. Remember to set `Linit` to `.false.` in the `settings.in` file if this file should be used.

2.7 wolfplot.in

This file can be added to the INPUT folder if one would like to calculate the energy of a configuration for different force field parameters. If this file is not present, the energy is calculated by using the parameters in the `forcefield.in` file. The `wolfplot.in` file has a fixed form (File 7). Using this file, the energy of a configuration for different cutoff radii and damping parameters can be calculated. The Ewald method is used once to calculate the electrostatic energy of the system and the Ewald parameters are defined by its precision. Results are written to the file `energy-box-1.dat` (and `energy-box-2.dat`) in the OUTPUT folder. Remember to set `LWolfPlot` to `.true.` and the number of boxes `Nbox` in the `settings.in` file. Furthermore, a `restart.in` file should be present in the INPUT folder.

2.8 umbrella.in

This file can be added to the INPUT folder if one would like to use a specific range of pressures or temperatures (β) for umbrella sampling [25]. By default, umbrella sampling calculates a range of pressures and temperatures based on typical values of the energy and volume of the system. In case this does not work as intended, the file `umbrella.in` can be used to set the ranges manually. The `umbrella.in` file has a fixed form (File 8).

```

Nbox      Temperature  Pressure   Reduced Units?
2          250          -1 bar     .false.

-----
Nproduction    Nequilibrat e Ninitialize
5E3           2E3           1E2

-----
Nconfiguration   Ndata       Naverage   Nrdf
1E2           1E1           1E0        0E0

-----
Init?    Weight?   Seed?    Seed
.true.   .false.   .false.   1669196305

-----
RDF_Molecule? RDF_Atom?  Insertions? WolfPlot?
.false.   .false.   .false.   .false.

-----
WL?    Fmod   Fred   Flatc  Linear?  Spline? dU/dL?
.true.  0.1    4.0    0.2    .false.   .false.   .false.

#####
----- Translation Moves -----
Normal    Pair      Cluster   Smart
35        0         0         0

----- Rotation Moves -----
Normal    Pair      Cluster   Smart
30        0         0         0

----- Volume Moves -----
Normal    Cluster
1          0

----- Intramolecular Moves -----
Bending   Torsion
0          0

----- Lambda Moves -----
Normal    GCMC
17        0

----- CFC Hybrid Moves -----
NVT/NPT   Gibbs     Reaction
0          0         0

----- Hybrid Move Switches -----
NVT/NPT   Gibbs     Reaction
0.0      1.0      0.3      0.7      0.0      1.0

```

File 1: Typical `settings.in` for a simulation in the Gibbs Ensemble. A negative number for the pressure means that the total volume is constant (*NVT* version of the Gibbs ensemble).

```

2 atomtype(s)
Atom Sigma      Epsilon     Charge   LJ? EL? Print
CH2 3.95000  46.00000  0.000000  T   F   C
CH3 3.75000  98.00000  0.000000  T   F   C
-----
1 bendingtype(s)
Bending    theta0      K      dtheta
TraPPE_C-C-C 114.00  62500.00  20.00
-----
1 torsiontype(s)
Torsion      A0      A1      A2      A3      A4      A5      dphi
TraPPE_X-CH2-CH2-X 0.000  355.030 -68.190  791.320           180.0
#####
Box 1      Box 2
Ideal_Gas      .false.      .false.
Cutoff_LJ_Energy  14.00      14.00
Method_EL_Energy  none      none
-----
LJ_Truncation      Tailcorrections
Mixing_Rules      Lorentz-Berthelot
-----
0 overrides
atom 1  atom 2  Sigma  Epsilon  Rmin
-----
```

File 2: `forcefield.in` for the simulation of (united atom) butane with the TraPPE force field [18]. Note that settings for calculation of electrostatic energies could be omitted since this model of butane has no charges.

```

Molecule      # Box 1      # Box 2
Butane        200          20
#####
1 Fractional Group
Type, Box, NinFrac MoleculeTypes and Name
GE      1      1      Butane      Butane-fractional
#####
0 Reaction(s)
#####
Box    Length    dVolume
1      30.000   270.0
2      30.000   270.0
-----
                           Box 1           Box 2
MolType     dtrans   drotation   dtrans   drotation
Butane      0.60     90.0       6.00     180.0
-----
                           Box 1
FracType   dlambda
1          0.2
#####
0 MolType Pair(s)
-----
Cluster Radius = 0.0

```

File 3: `topology.in` for a simulation of butane in the Gibbs Ensemble.

```

Molecule # Box 1
N2      120
H2      360
NH3      0
#####
Partition Functions
N2      0.0
H2      0.0
NH3      0.0
#####
0 Fractional Group(s)
Type, Box, NinFrac MoleculeTypes and Name
#####
1 Reaction
-----
3 MolTypes in Reaction in Box 1
MolType   Nstoi
N2        -1
H2        -3
NH3        2
#####
Box    Length   dVolume
 1     30.000   270.0
-----
          Box 1
MolType   dtrans  drotation
N2        0.60    90.0
H2        0.60    90.0
NH3        0.60    90.0
-----
          Box 1
Fractional  dlambda
 1          0.2
#####
0 MolType Pair(s)
-----
Cluster Radius = 0.0

```

File 4: `topology.in` for a simulation of the reaction $\text{N}_2 + 3\text{H}_2 \rightleftharpoons 2\text{NH}_3$.

```

### g/mol
-----
### atom(s)
Atom X-pos Y-pos Z-pos
. .
. .
. .

Atom X-pos Y-pos Z-pos
-----
### bond(s)
Atom 1 Atom 2
. .
. .

Atom 1 Atom 2
-----
### bending(s)
Bending-name Atom 1 Atom 2 Atom 3 Lambda_Scaling?
. .
. .

Bending-name Atom 1 Atom 2 Atom 3 Lambda_Scaling?
-----
### torsion(s)
Torsion-name Atom 1 Atom 2 Atom 3 Atom 4 Lambda_Scaling?
. .
. .

Torsion-name Atom 1 Atom 2 Atom 3 Atom 4 Lambda_Scaling?
-----
### intramolecular interaction(s)
Atom 1 Atom 2 ScaleLJ6 ScaleLJ12 ScaleEL Lambda_Scaling LJ and EL?
. .
. .

Atom 1 Atom 2 ScaleLJ6 ScaleLJ12 ScaleEL Lambda_Scaling LJ and EL?

```

File 5: Molecule definition template.

```

58.12 g/mol
-----
4 atoms
CH3  0.000000000  0.000000000  0.000000000
CH2  1.291552675  0.838744114  0.000000000
CH2  2.583105349  0.000000000  0.000000000
CH3  3.874658024  0.838744114  0.000000000
-----
3 bonds
 1 2
 2 3
 3 4
-----
2 bendings
TraPPE_C-C-C    1  2  3   T
TraPPE_C-C-C    2  3  4   T
-----
1 torsion
TraPPE_X-CH2-CH2-X 1  2  3  4   T
-----
0 intramolecular interaction(s)

```

File 6: Butane molecule definition.

Parameter	Box 1:	Start	End	Step	Box 2:	Start	End	Step
Wolf_CutOff		10.0	17.0	1.0		10.0	30.0	5.0
Wolf_Alpha		0.00	0.2	0.01		0.00	0.1	0.01

Ewald_Precision 1E-6

File 7: Typical `wolfplot.in` for calculating the electrostatic energy of a configuration with different Wolf parameters.

Parameter	Box 1:	Delta	Nbins	Box 2:	Delta	Nbins
Pressure		0.05	10		0.05	10
Beta		0.05	10		0.05	10

File 8: Typical `umbrella.in` for setting ranges for umbrella sampling. In this example file, 10 bins are used for the pressure range in Box 1 and they are each 0.05 distance away from each other. This means that 5 bins correspond to pressures lower than the defined pressure for the simulation and 5 bins correspond to pressures higher. For example, if the pressure would be equal to 2, we have bins for the following values: 1.75, 1.80, 1.85, 1.90, 1.95, 2, 2.05, 2.10, 2.15, 2.20 and 2.25. Note that the defined pressure itself is always assigned a bin too.

Chapter 3

OUTPUT

All output files, except `sim.log`, are collected in the `OUTPUT` folder. There are four types of output files: `.dat` files can be used for plotting, `.info` files contain simulation information and are mostly used for debugging, `.out` files can be used as input for new simulations and `.xyz` files can be used to visualize the system.

3.1 sim.log

This file is only created if Brick-CFCMC is run without the `-t/-terminal` option, otherwise the output is written to the terminal.

- **Energy Starting Configuration:** The energies/ k_B for all interactions of the initial system. Units: energies in K.
- **Relative Energy Drifts:** The relative energy drifts of all interaction terms. This information is also written to the file `energy_drifts.info`. Unitless.
- **Trial Move Acceptance Rates:** The acceptance rates of all trial moves. This information is also written to the file `trial_moves.info` Unitless.
- **Averages:** The average system properties. Units: density in kg/m³ or number density if reduced units are used, volume in Å³ or reduced units, other properties are unitless.
- **Properties from CFCMC:** The properties that have been calculated using the CFCMC technique. Units: chemical potential/ k_B in K, partial molar enthalpy/ k_B in K, partial molar volume in Å³ or reduced units, other properties are unitless.
- **Properties from CFC-GE:** The chemical potentials that have been calculated using the CFCMC technique for the Gibbs Ensemble (note that these can be used as an indication and one should use probes instead). Units: K.
- **Properties from CFC-RxMC:** The chemical potentials, fugacity coefficients that have been calculated using the CFCMC technique for the Reaction Ensemble (note that these can be used as an indication and one should use probes instead), and the equilibrium constants. Units: chemical potential/ k_B in K.

- **Particle Insertion Methods:** The chemical potentials, partial molar enthalpies and partial molar volumes from Widom's test particle insertion methods. Units: chemical potential/ k_B in K, partial molar enthalpy/ k_B in K and partial molar volume in \AA^3 or reduced units.
- **Properties from Umbrella Sampling:** The partial molar enthalpies and partial molar volumes calculated from umbrella sampling. Units: chemical potential in K, partial molar enthalpy in K and partial molar volume in \AA^3 or reduced units.
- **Energy Starting Configuration:** The average energies of the system. Units: energies in K.

3.2 System Properties

- **density_vs_cycle.dat:** The density of molecule types in each simulation box as a function of the number of MC cycles. Units: kg/m^3 (or number density if reduced units are used)
- **energy_vs_cycle.dat:** energies/ k_B in each simulation box as a function of the number of MC cycles. Units: K .
- **number_of_molecules_vs_cycle.dat:** The number of molecules of each molecule type in each simulation box as a function of the number of MC cycles. Unitless.
- **volume_vs_cycle.dat:** The volume of each simulation box as a function of the number of MC cycles. Units: \AA^3 (or reduced units if selected).

3.3 Averages

- **av_density.dat:** The average density of molecule types in each simulation box as a function of the number of MC cycles. Units: kg/m^3 (or number density if reduced units are used)
- **av_energy.dat:** The average energies/ k_B in each simulation box as a function of the number of MC cycles. Units: K .
- **av_number_of_molecules.dat:** The average number of molecules of each molecule type in each simulation box as a function of the number of MC cycles. Unitless.
- **av_volume.dat:** The average volume of each simulation box as a function of the number of MC cycles. Units: \AA^3 (or reduced units if selected).

3.4 Configurations

Configuration files are placed in the subdirectory CONFIGURATIONS. These files can be opened with software like Jmol.

- **molecule.xyz:** The configurations of all molecules of a certain molecule type in the system during the simulation.

- **molecule_initial.xyz**: The initial configuration of all molecules of a certain molecule type in the system.
- **molecule_final.xyz**: The final configuration of all molecules of a certain molecule type in the system.
- **ALL.xyz**: The configuration of all molecules in the system.
- **ALL_initial.xyz**: The initial configuration of all molecules in the system.
- **ALL_final.xyz**: The final configuration of all molecules in the system.

3.5 Radial Distribution Functions

The files containing Radial Distribution Functions are placed in the subdirectory RDF.

- **atom-atom_A-B.dat**: The RDF of atom type A and atom type B.
- **molecule-molecule_A-B.dat**: The RDF of molecule type A and molecule type B.

3.6 Umbrella Sampling

The files containing results from umbrella sampling [25] are placed in the subdirectory UMBRELLA_SAMPLING. Chemical potentials are calculated only for Fractional Groups and a separate file is generated for each of them.

- **ChemicalPotential_vs_Beta-#.dat**: Chemical potential/ k_B for different values of β . Units: K .
- **ChemicalPotential_vs_Pressure-#.dat**: Chemical potential/ k_B for different values of pressure. Units: K .
- **Density_vs_Beta.dat**: Average density for different values of β . Units: kg/m^3 (or number density if reduced units are used).
- **Density_vs_Pressure.dat**: Average density for different values of pressure. Units: kg/m^3 (or number density if reduced units are used).

3.7 CFC

Properties calculated with the CFC method [2] are placed in the subdirectory CFC. For each fractional, separate files are generated (numbered in order of the definition in the `topology.in` file). Each of the .dat-files each consists of 10 blocks of values. The first block contains the resulting averages after 10% of the production cycles, the second block after 20%, etc. The final averages (overall results) are in the last block (use (i)ndex in `gnuplot` to plot the appropriate block of data).

- **1overN_vs_Lambda-#.dat**: Average $1/(N + N_{\text{frac}})$ as a function of λ . Unitless.

- **1overV_vs_Lambda-#.dat**: Average inverse volume as a function of λ . Units: $1/\text{\AA}^3$ (or reduced units if selected).
- **dU_dLambda.out**: Average $(\partial U / \partial \lambda) / k_B$ as a function of λ (2nd column) and standard deviation (3rd column). Units: K.
- **H_vs_Lambda-#.dat**: Average enthalpy/ k_B of the system as a function of λ . Units: K.
- **HoverV_vs_Lambda-#.dat**: Average enthalpy/ k_B divided by the volume of the system as a function of λ . Units: $\text{K}/\text{\AA}^3$ (or reduced units if selected).
- **Hsquared_vs_Lambda-#.dat**: Average squared enthalpy/ k_B of the system as a function of λ . Units: K^2 .
- **HsquaredoverV_vs_Lambda-#.dat**: Average squared enthalpy/ k_B divided by the volume of the system as a function of λ . Units: $\text{K}^2/\text{\AA}^3$ (or reduced units if selected).
- **plambda-#.dat**: The λ -probability distribution (biased and observed). Unitless.
- **plambda.out**: The final λ -probability distribution (biased and observed). It can be used for the iterative scheme. Unitless.
- **Utotal_vs_Lambda-#.dat**: Average total energy of the system as a function of λ . Units: K.
- **dUdl_vs_cycle_for_alchemlyb#.dat**: Instantaneous value of $(\partial U / \partial \lambda) / k_B$ as a function of the number of MC cycles (1st column) and λ (2nd column). In this file, λ (2nd column) values are the centers of the λ bins. Units: K.
- **V_vs_Lambda-#.dat**: Average volume of the system as a function of λ . Units: \AA^3 .

Note that depending on the type of the fractional not all of these files will be generated.

3.8 Weight function and Restart

Weight functions are placed in the subdirectory **WEIGHTFUNCTIONS**. For each fractional a separate file is generated (numbered in order of the definition in the **topology.in** file).

- **restart.out**: A restart file which can be used as input for a new simulation.
- **weightfunction.out**: Final weight function, which is used in the production phase of the simulation. It can be used as input for a new simulation.
- **weightfunction-#.dat**: Every time a weight function meets the flatness criterion in the Wang Landau algorithm [17], the weight function is added to this file, i.e. one can see from this file how the weight function is being constructed.

3.9 CFCMC trial moves

The following files contain two blocks. The first block contains the acceptance ratio as a function of λ . The second block contains the total number of attempted trial moves and accepted trial moves.

- **NVPT_change_move.dat**: Acceptance ratios for the NVT/NPT identity change trial move as a function of λ for each fractional type.
- **NVPT_swap_move.dat**: Acceptance ratios for the NVT/NPT swap trial move as a function of $\bar{\lambda}$ for each fractional type.
- **GE_change_move.dat**: Acceptance ratios for the Gibbs Ensemble identity change trial move as a function of λ for each fractional type.
- **GE_swap_move.dat**: Acceptance ratios for the Gibbs Ensemble swap trial move as a function of $\bar{\lambda}$ for each fractional type.
- **RXMC_change_move.dat**: Acceptance ratios for the Reaction Ensemble identity change trial move as a function of λ for each fractional type.
- **RXMC_swap_move.dat**: Acceptance ratios for the Reaction Ensemble swap trial move as a function of $\bar{\lambda}$ for each fractional type.

3.10 Simulation Information

- **energy_drifts.info**: Information about energy drifts in the system.
- **interactions.info**: Information about the interactions in the system.
- **RNG.info**: Information about the seed for the random number generator (Mersenne twister [14]).
- **trial_moves.info**: Information about the number of trials and acceptances for the MC trial moves used in the simulation.

Chapter 4

Commands

There are options/actions that you can call via the terminal. All commands are preceded by the keyword **brick** (for example '**brick compile**' would result in compiling the code and tools).

4.1 Overview

In Brick-CFCMC the following commands can be used:

command	options	action
calculate	Ewald-parameters	calculate Ewald parameters
	fugacity-coefficients	calculate fugacity coefficients
	partition-functions	calculate partition functions
compile	-o/--optimize	compile Brick-CFCMC with optimizations
	-d/--debug	compile Brick-CFCMC with debug options
	-i/--intel/--ifort	compile Brick-CFCMC with the Intel Fortran compiler
	-g/--gnufortran/--gfortran	compile Brick-CFCMC with GNU Fortran compiler
	-t/--tools	compile the Brick-CFCMC tools (using ifort)
compile		execute all of the above
integrate	dUdλ	integrate $\partial U / \partial \lambda$
new	<name>	create a new simulation directory

The following commands do not take any arguments:

command	action
<code>input</code>	create input for a new simulation
<code>iterative-scheme</code>	apply the iterative scheme to a weight function
<code>pack</code>	pack input and output to an archive
<code>source</code>	go to the source code
<code>smoothen-weightfunction</code>	fit a spline through a weight function
<code>update</code>	update Brick-CFCMC
<code>wolfplot</code>	generate a 'wolf plot' with gnuplot

4.2 Detailed description

Complete command	Description
<code>brick</code>	The keyword without any arguments will bring you to the main directory of Brick: it executes the command <code>cd \${BRICK_DIR}</code> .
<code>brick calculate Ewald-parameters</code>	This will run a small program to calculate the recommended Ewald parameters depending on the box size. Note that this command can take a second argument which is the box size for which one wants to calculate the parameters.
<code>brick calculate fugacity-coefficients</code>	This will run a small program to calculate fugacity coefficients for one or more molecule types. Fugacity coefficients are calculated with the Peng-Robinson equation of state.
<code>brick calculate partition-functions</code>	This will run a small program to calculate the natural logarithm of the partition functions for one or more molecule types.
<code>brick compile</code>	This will execute the command <code>Brick compile</code> four times with different options: <code>-i -o</code> , <code>-i -d</code> , <code>-g -o</code> and <code>-g -d</code> .
<code>brick input</code>	This will start the input creator which creates the <code>topology.in</code> and <code>forcefield.in</code> files
<code>brick integrate dUdl</code>	This will run a small program to numerically integrate $\partial U / \partial \lambda$ using the trapezoidal rule to obtain the excess chemical potential.
<code>brick iterative-scheme</code>	This will apply the iterative scheme to a weight function. By default it uses the <code>weightfunction.out</code> and <code>plambda.out</code> files.
<code>brick new <name></code>	This will create a new directory to start a simulation: it creates the directory, creates an INPUT folder with <code>input</code> file and places a <code>run</code> script in the folder.
<code>brick pack</code>	This will create an archive with the INPUT and OUTPUT folders and the file <code>sim.log</code> , it will leave out the <code>configurations.xyz</code> file. This archive contains all information that is needed for debugging.

<code>brick smoothen-weightfunction</code>	This will start the smoothen-weightfunction tool which will attempt to create a smoother weight function. It starts by rebinning the weightfunction and then fits a spline through this rebinned function. The resulting spline is then discretized to be used in a new simulation.
<code>brick source</code>	This will bring you to the main source code: it executes the command: <code>cd \${Brick_DIR}\SOURCE</code>
<code>brick update</code>	This will update your version of Brick-CFCMC by executing the command: <code>git pull</code> in the <code> \${BRICK_DIR}</code> directory
<code>brick wolfplot</code>	This will generate a 'Wolf Plot' with gnuplot. First, the program has to be run with <code>LWolfPlot</code> set to <code>.true.</code>

Chapter 5

Energy

In this section we describe how the energy of the system is calculated.

5.1 Intermolecular Lennard-Jones Interactions

The well-known Lennard-Jones potential has the form:

$$U_{\text{LJ}}(r_{ij}) = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (5.1)$$

where ϵ_{ij} and σ_{ij} are the LJ-parameters between atom i and j . The potential is truncated at a certain cutoff distance R_c .

5.1.1 Mixing Rules

To calculate the LJ-parameters σ_{ij} and ϵ_{ij} between two atom types from the single component parameter σ_i and ϵ_i mixing rules (combining rules) are used. See Section 2.2.5 for details on how to select mixing rules in Brick-CFCMC.

Lorentz-Berthelot

Most commonly, Lorentz-Berthelot mixing rules [26] are used:

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \quad (5.2)$$

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j} \quad (5.3)$$

Jorgensen (OPLS)

A different set of mixing rules are the Jorgensen (OPLS) mixing rules [27]:

$$\sigma_{ij} = \sqrt{\sigma_i \sigma_j} \quad (5.4)$$

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j} \quad (5.5)$$

Minimum Distance

In order to avoid singularities when $r \rightarrow 0 \text{ \AA}$ (i.e. unrealistic large energies) a minimum cutoff $r_{\min,ij}$ is set. This minimum value is chosen dependent on the LJ-parameter σ_{ij} :

$$r_{\min,ij} = \frac{1}{2}\sigma_{ij} \quad (5.6)$$

Whenever two atoms (except for Fractionals) are closer than this distance the trial move gets rejected immediately. For Fractionals a different approach is used (see Section 5.1.4).

5.1.2 Shifting the Potential

There are two common ways to account for the error introduced by truncating the potential. One of them is to shift the potential. Shifting the LJ potential at the cutoff makes the potential continuous at R_c :

$$U_{\text{LJ}}^{\text{shifted}}(r_{ij}) = \begin{cases} U_{\text{LJ}}(r_{ij}) - U_{\text{LJ}}(R_c) & r_{ij} \leq R_c \\ 0 & r_{ij} > R_c \end{cases} \quad (5.7)$$

5.1.3 Analytic Tail Corrections

A second way of accounting for the errors introduced by truncation is applying analytic tail corrections. They can be interpreted as an average energy contribution of molecules beyond the cutoff. The tail corrections are given by [26]:

$$U_{\text{LJ}}^{\text{tailcorrection}} = \frac{1}{2} \sum_{i,j} \frac{16\pi N_i N_j \epsilon_{ij}}{V} \left(\frac{\sigma_{ij}^{12}}{9R_c^9} - \frac{\sigma_{ij}^6}{3R_c^3} \right) \quad (5.8)$$

where the sum ranges over all atom types in the system, N_i is the number of atoms of type i (excluding the fractional molecules), and V is the volume of the simulation box. The factor $\frac{1}{2}$ accounts for double counting interactions. See Section 2.2.5 for details on how to define the details for truncating in Brick-CFCMC.

5.1.4 Scaling for Fractional Molecules

The intermolecular LJ interactions of the fractional molecules in the CFC method are scaled according to [28, 29]:

$$U_{\text{LJ}}(r_{ij}, \lambda_{\text{LJ}}) = 4\lambda_{\text{LJ}}\epsilon_{ij} \left[\frac{1}{\left(\alpha(1 - \lambda_{\text{LJ}}) + \left(\frac{r_{ij}}{\sigma_{ij}} \right)^c \right)^{\frac{12}{c}}} - \frac{1}{\left(\alpha(1 - \lambda_{\text{LJ}}) + \left(\frac{r_{ij}}{\sigma_{ij}} \right)^c \right)^{\frac{6}{c}}} \right] \quad (5.9)$$

Where λ_{LJ} is the fractional scaling parameter for LJ interactions. This parameter is a function of the overall fractional parameter λ . Shifting this potential is straightforward. There are multiple ways of adding the fractional interaction to the tail correction, we do this by substituting $N_i \rightarrow N_i + \lambda_{\text{LJ}}$ in Eq. (5.8). This ensures that for $\lambda_{\text{LJ}} = 0$, the fractional molecule does not contribute to the tail correction, and for $\lambda_{\text{LJ}} = 1$ the contribution is equal to that of a normal molecule in the system. The term $\alpha(1 - \lambda_{\text{LJ}})$ in the denominators prevents singularities from happening when $\lambda_{\text{LJ}} \rightarrow 0$.

5.2 Intermolecular Electrostatic Interactions

Charges interact via the well-known Coulomb potential:

$$U_{\text{Coulomb}}(r_{ij}) = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \quad (5.10)$$

where q_i and q_j are charges and ϵ_0 is the permittivity of free space. Since this is a long range interaction, one needs to correct for the error that is introduced when truncating the potential at R_c . The Ewald summation method [23] is often used, however, there are alternatives such as the Wolf method [21]. In the following sections we leave out the factor $\frac{1}{4\pi\epsilon_0}$ for simplicity. Here we only consider the intermolecular interactions. Intramolecular interactions are handled separately and are described in Section 5.4.

5.2.1 Ewald Method

The Ewald summation [23] splits the electrostatic interaction in a short-range and a long-range part. The short-range part can then be calculated directly and for the long-range part a Fourier Transform is used. The total intermolecular electrostatic energy is then given by:

$$\begin{aligned} U_{\text{electrostatic}}^{\text{Ewald}} = & \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{j=1}^{N_m} \sum_{\substack{b=1 \\ j \neq i \\ r_{iajb} < R_c}}^{N_a^j} q_{ia} q_{jb} \frac{\operatorname{erfc}(\alpha r_{iajb})}{r_{iajb}} \\ & + \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{\substack{b=1 \\ b \neq a \\ r_{iaib} < R_c}}^{N_a^j} q_{ia} q_{ib} \frac{\operatorname{erfc}(\alpha r_{iaib})}{r_{iaib}} \\ & - \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{b=1}^{N_a^j} q_{ia} q_{ib} \frac{1}{r_{iaib}} \\ & + \frac{1}{2V} \sum_{\vec{k} \neq 0} \frac{4\pi}{k^2} \left| \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} q_{ia} \exp[i\vec{k} \cdot \vec{r}_{ia}] \right|^2 \exp\left[-\frac{k^2}{4\alpha^2}\right] \\ & - \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} q_{ia}^2 \end{aligned} \quad (5.11)$$

where N_m is the number of molecules, N_a^i is the number of atoms in molecule i and j , q_{ia} is the partial charge of atom a in molecule i , erfc is the complementary error function, α is a damping parameter, $r_{iajb} = |\vec{r}_{ia} - \vec{r}_{jb}|$ is the distance between atom a in molecule i and atom b in molecule j , R_c is the cutoff radius. The sum in the second term on the right hand side ranges over vectors $\vec{k} = \frac{2\pi}{L}(n_x, n_y, n_z)$ with $L = V^{\frac{1}{3}}$ (the box length) and integers $n_x, n_y, n_z \in \mathbb{N}$. Since this is a converging sum we can truncate it at a certain maximum vector k_{\max} (or n_{\max} such that $n_x, n_y, n_z \leq n_{\max}$). In Brick-CFCMC, the Fourier part of the Ewald summation is stored in such a way that when a change is applied to a single molecule, only terms related to this single molecule have to be recomputed [30].

Choosing Ewald Parameters

Based on the aimed relative precision (δ), that one wants to achieve for calculating electrostatic interactions, it is possible to determine the values of α , R_c and n_{\max} (k_{\max}) [31]:

$$R_c = 0.4L \quad (5.12)$$

$$\alpha = \sqrt{|\ln(\delta R_c)|}/R_c \quad (5.13)$$

$$n_{\max} = \left[\frac{1}{4} + \frac{L\alpha\omega}{\pi} \right] \quad (5.14)$$

where

$$\eta = \sqrt{|\ln(\delta R_c)|} \quad (5.15)$$

$$\omega = \sqrt{|\ln(4\eta^2\delta\alpha^2R_c)|} \quad (5.16)$$

where L is the box length and $[x]$ rounds the argument x to the nearest integer.

5.2.2 Wolf Method

In the Wolf summation method [21] the following expression is used to calculate intermolecular electrostatic interactions:

$$\begin{aligned} U_{\text{electrostatic}}^{\text{Wolf}} = & \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{j=1}^{N_m} \sum_{b=1}^{N_a^j} q_{ia} q_{jb} \left(\frac{\operatorname{erfc}(\alpha r_{iajb})}{r_{iajb}} - \frac{\operatorname{erfc}(\alpha R_c)}{R_c} \right) \\ & \quad \underset{j \neq i}{\underset{r_{iajb} < R_c}{}} \\ & + \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{b=1}^{N_a^j} q_{ia} q_{ib} \left(\frac{\operatorname{erfc}(\alpha r_{iaib})}{r_{iaib}} - \frac{\operatorname{erfc}(\alpha R_c)}{R_c} \right) \\ & \quad \underset{b \neq a}{\underset{r_{iaib} < R_c}{}} \\ & - \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{b=1}^{N_a^j} q_{ia} q_{ib} \frac{1}{r_{iaib}} \\ & - \left(\frac{\operatorname{erfc}(\alpha R_c)}{2R_c} + \frac{\alpha}{\sqrt{\pi}} \right) \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} q_{ia}^2 \end{aligned} \quad (5.17)$$

where we refer to Sec 5.2.1 for the meaning of all symbols. Note that α is again a damping parameter but **different** from the one used in the Ewald method. Typically, the value of α needs to be larger in a liquid system than in a gas, and the value of R_c is smaller in a liquid than in a gas. This is related to the fact that in a liquid the effective screening is larger and the interaction is more short-ranged.

Choosing Wolf Parameters

In contrast to the Ewald method, there is no direct criterion for determining the values of α and R_c in the Wolf method. One has to follow the following procedure [32]:

- Run a short *NVT* simulation at a density close to equilibrium (obtained from literature for example) above the critical temperature, so that no phase separation occurs in the simulation box. Choose a small value of α and a large R_c (preferably half the box size).
- Calculate the energy of the final configuration with many different Wolf parameters:

Copy `restart.out` to the INPUT folder (as `restart.in`)

Set `LWolfPlot` to `.true.` in `settings.in`

Copy the template `wolfplot.example` to `wolfplot.in` in INPUT

Edit the file `wolfplot.in` to set ranges for the Wolf parameters

Run the program

- Plot the electrostatic energy calculated with the Ewald summation (which we consider as the exact solution) and the Wolf method for different parameters (execute `Brick wolfplot`)

Typical resulting plots for a liquid and vapor can be found in Figures 5.1 and 5.2 respectively. Figure 5.1 also clearly shows the effect of a damping parameter that is too small: for $\alpha \rightarrow 0 \text{ \AA}$ and large R_c the lack of screening in the cutoff spheres result in large energy differences. From those plots, the optimal values for α and R_c can be determined i.e. the parameters that give an accurate result compared to the Ewald summation, choosing R_c as small as possible.

Initial Guess

Possible suitable initial guesses for the Wolf parameters are:

$$\alpha = 0.12 \text{ \AA}^{-1} \text{ and } R_c = 14 \text{ \AA} \text{ for a liquid} \quad (5.18)$$

$$\alpha = 0.04 \text{ \AA}^{-1} \text{ and } R_c = 25 \text{ \AA} \text{ for a gas} \quad (5.19)$$

The applicability of the Wolf method has been demonstrated in the study of vapor-liquid equilibria of hydrogen sulfide, methanol, carbon dioxide, and xylenes [32, 4, 5] and computation of partial molar properties of water-methanol mixtures [33].

5.2.3 Fennell-Gezelter (Damped Shifted Force Potential)

Fennell and Gezelter modified the Wolf method [22] to obtain a continuous force at the cutoff radius:

$$\begin{aligned}
U_{\text{electrostatic}}^{\text{Fennell-Gezelter}} = & \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{j=1}^{N_m} \sum_{\substack{b=1 \\ j \neq i \\ r_{iajb} < R_c}}^{N_a^j} q_{ia} q_{jb} \left[\frac{\operatorname{erfc}(\alpha r_{iajb})}{r_{iajb}} - \frac{\operatorname{erfc}(\alpha R_c)}{R_c} \right. \\
& + \left. \left(\frac{\operatorname{erfc}(\alpha R_c)}{R_c^2} + \frac{2\alpha}{\sqrt{\pi}} \frac{\exp[-\alpha^2 R_c^2]}{R_c} \right) (r_{iajb} - R_c) \right] \\
& + \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{\substack{b=1 \\ b \neq a \\ r_{iaib} < R_c}}^{N_a^j} q_{ia} q_{ib} \left(\frac{\operatorname{erfc}(\alpha r_{iaib})}{r_{iaib}} - \frac{\operatorname{erfc}(\alpha R_c)}{R_c} \right) \\
& - \frac{1}{2} \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} \sum_{b=1}^{N_a^j} q_{ia} q_{ib} \frac{1}{r_{iaib}} \\
& - \left(\frac{\operatorname{erfc}(\alpha R_c)}{2R_c} + \frac{\alpha}{\sqrt{\pi}} \right) \sum_{i=1}^{N_m} \sum_{a=1}^{N_a^i} q_{ia}^2
\end{aligned} \tag{5.20}$$

Minimum Distance

To avoid singularities the same minimum cutoff as for LJ-interactions is taken.¹

$$r_{\min,ij} = \frac{1}{2} \sigma_{ij} \tag{5.21}$$

Whenever two atoms (except for Fractionals) are closer than this distance the trial move gets rejected immediately. Again, for Fractionals a different approach is used (see 5.2.4).

5.2.4 Scaling for Fractional Molecules

The electrostatic interaction of a fractional molecule is scaled by substituting $q_{ia} \rightarrow \lambda_{\text{el}} q_{ia}$ for the atoms of the Fractional Molecule and $r_{iajb} \rightarrow r_{iajb} + \beta_{\text{el}} (1 - \lambda_{\text{el}})^2$ (or r_{iaib}) in the above equations. Here, λ_{el} is the fractional scaling parameter for electrostatic interactions and it is a function of the overall fractional parameter λ . The term $\beta_{\text{el}} (1 - \lambda_{\text{el}})^2$ prevents singularities for very small values of r_{iajb} . The value of β_{el} is chosen as $\frac{1}{2}$ Å.

¹This assumes that every atom type has some LJ interaction.

5.3 Intramolecular Lennard-Jones Interactions

As described earlier in Section 2.4 the intramolecular Lennard-Jones interactions use scaling parameters. The potential used to calculate these interactions is as follows:

$$U_{\text{LJ}}^{\text{intramolecular}}(r_{ij}) = 4\epsilon_{ij} \left[\alpha_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \beta_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (5.22)$$

where α_{ij} and β_{ij} are the scaling parameters. For intramolecular interactions there is no truncation, i.e. no cutoff is used.

5.4 Intramolecular Electrostatic Interactions

Intramolecular electrostatic interactions in a molecule can be scaled. We use the following form to calculate the energy:

$$U_{\text{electrostatic}}^{\text{intramolecular}}(r_{ij}) = \gamma_{ij} \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \quad (5.23)$$

where γ_{ij} is the scaling parameter. For intramolecular interactions there is no truncation, i.e. no cutoff is used.

5.5 Bending

A commonly used form of bending potential is the quadratic one:

$$U_{\text{bending}}(\theta) = \frac{K}{2} (\theta - \theta_0)^2 \quad (5.24)$$

where K is the bond bending constant, θ is the angle between two bonds and θ_0 is the equilibrium angle.

5.6 Torsion

Torsion potentials come in many forms and definitions. The torsion potential is a function of the torsion angle

$$U_{\text{torsion}} = U_{\text{torsion}}(\phi) \quad (5.25)$$

More important, in general this is an even and periodic function and it can therefore be written as a series of cosines:

$$U_{\text{torsion}}(\phi) = \sum_{i=0}^{\infty} p_i \cos^i \phi \quad (5.26)$$

Consider for example the TraPPE definition:

$$U_{\text{torsion}}(\phi) = c_0 + c_1[1 + \cos \phi] + c_2[1 - \cos 2\phi] + c_3[1 + \cos 3\phi] \quad (5.27)$$

which can be written as a series of cosines with:

$$\begin{aligned} p_0 &= c_0 + c_1 + 2c_2 + c_3 \\ p_1 &= c_1 - 3c_3 \\ p_2 &= -2c_2 \\ p_3 &= 4c_3 \\ p_i &= 0 \text{ for } i > 4 \end{aligned} \tag{5.28}$$

This way, most torsion potentials can be rewritten. Furthermore, this is also used in the algorithms to calculate the energy because it is computationally cheaper. We limit the series to six terms:

$$U_{\text{torsion}}(\phi) = \sum_{i=0}^5 p_i \cos^i \phi \tag{5.29}$$

5.7 Efficient Scaling of Fractional Interactions

In the above sections we introduced two different scaling parameters for different interactions: λ_{LJ} , λ_{el} . These parameters are functions of the overall fractional parameter λ and are restricted to the following conditions:

$$\begin{cases} \lim_{\lambda \downarrow 0} \lambda_{\text{LJ/el}}(\lambda) = 0 \\ \lim_{\lambda \uparrow 1} \lambda_{\text{LJ/el}}(\lambda) = 1 \end{cases} \tag{5.30}$$

such that there are no interactions (the fractional molecules behave like an ideal gas) when $\lambda \downarrow 0$ and the fractional molecules have normal interactions when $\lambda \uparrow 1$. Because those are the only restrictions we are free to choose how the functions look like on the rest of the λ -interval $(0, 1)$. We choose to first build up the LJ interactions and then the electrostatics. We do this by introducing a parameter λ_s which cuts the interval $(0, 1)$ in two. On the first part λ_{LJ} is gradually increased such that

$$\begin{cases} \lim_{\lambda \downarrow 0} \lambda_{\text{LJ}}(\lambda) = 0 \\ \lim_{\lambda \uparrow \lambda_s} \lambda_{\text{LJ}}(\lambda) = 1 \end{cases} \tag{5.31}$$

Then, on the second part we start building up the electrostatic interactions such that

$$\begin{cases} \lim_{\lambda \downarrow \lambda_s} \lambda_{\text{el}}(\lambda) = 0 \\ \lim_{\lambda \uparrow 1} \lambda_{\text{el}}(\lambda) = 1 \end{cases} \tag{5.32}$$

For more efficient sampling of properties that depend on the fractional parameter such as chemical potentials (Sections 6.4.3 and 6.6.3) we use the first and last bins for sampling more efficient [34]. Instead of starting to scale the interactions immediately from $\lambda = 0$ we start a bit later at $\lambda = \frac{1}{N}$ so that we have $\lambda_{\text{LJ/el}} = 0$ for $\lambda < \frac{1}{N}$. A similar thing is happening at $\lambda = \frac{N-1}{N}$ so that we have $\lambda_{\text{LJ/el}} = 0$ for $\lambda > \frac{N-1}{N}$. This way we can use the first and last bin to sample the probabilities of λ

truly being 0 (ideal gas) and 1 (whole molecule), which improves the accuracy of the calculation of partial molar properties.

In case thermodynamic integration is used, the scaling of Lennard-Jones and electrostatic interactions is slightly different [7] as we need to develop a scaling scheme that uses continuous functions λ_{LJ} and λ_{el} so that the integration of $\langle \partial U / \partial \lambda \rangle$ can be performed. Both λ_{LJ} and λ_{el} and the derivatives of these functions with respect to λ are required to be continuous functions. The following equations provide the used expressions for λ_{LJ} and λ_{el} , respectively:

$$\lambda_{\text{LJ}} = \begin{cases} \frac{20}{9}\lambda & 0.0 < \lambda < 0.4, \\ 1 - \frac{100}{9}(\lambda - \frac{1}{2})^2 & 0.4 < \lambda < 0.5, \\ 1 & 0.5 < \lambda < 1.0 \end{cases} \quad (5.33)$$

$$\lambda_{\text{el}} = \begin{cases} 0 & 0.0 < \lambda < 0.5, \\ \frac{100}{9}(\lambda - \frac{1}{2})^2 & 0.5 < \lambda < 0.6, \\ \frac{-11}{9} + \frac{20}{9}\lambda & 0.6 < \lambda < 1.0 \end{cases} \quad (5.34)$$

This is shown graphically in Fig. 5.3. With this scaling scheme, electrostatic interactions are not "switched on" before the LJ interactions are at full strength ($\lambda_{\text{LJ}} = 1$). This is chosen to avoid any overlap between the atoms of the fractional group and other atoms. In this way, we protect the electrostatic interaction sites using the LJ interactions in order to avoid atomic overlaps. The value of $\frac{\partial U}{\partial \lambda}$ is computed by using the chain rule:

$$\frac{\partial U}{\partial \lambda} = \frac{\partial U_{\text{LJ}}}{\partial \lambda_{\text{LJ}}} \frac{\partial \lambda_{\text{LJ}}}{\partial \lambda} + \frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}} \frac{\partial \lambda_{\text{el}}}{\partial \lambda} \quad (5.35)$$

With these definitions, $\frac{\partial U}{\partial \lambda} = 0$ at $\lambda = 0.5$. The terms $\frac{\partial U_{\text{LJ}}}{\partial \lambda_{\text{LJ}}}$ and $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$ are computed after every MC trial move and we keep track of these quantities after every MC trial move. This bookkeeping is implemented to avoid any additional computational cost to the simulations.

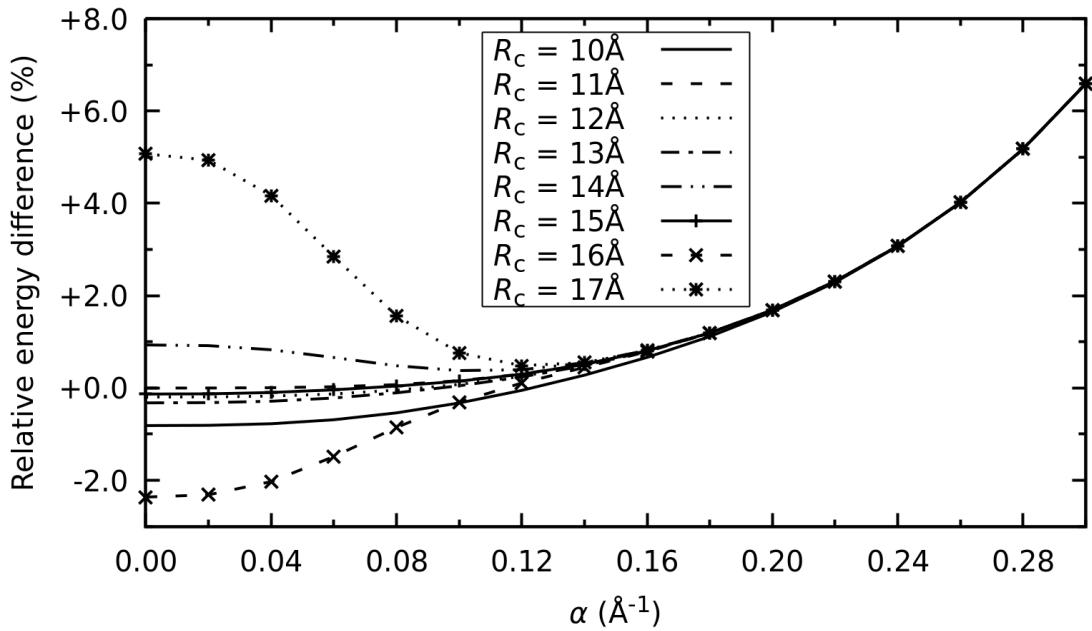


Figure 5.1: Relative difference in electrostatic energy between the Wolf method and Ewald summation for different Wolf parameters R_c as a function of α . The parameters for the Ewald summation are chosen such that a relative precision of 10^{-6} is achieved. The energy is calculated for methanol at a (typical liquid) density of 692 kg/m^3 at 600K using the force field from [35]. The optimal value of α is in the range from 0.10 \AA^{-1} to 0.14 \AA^{-1} .

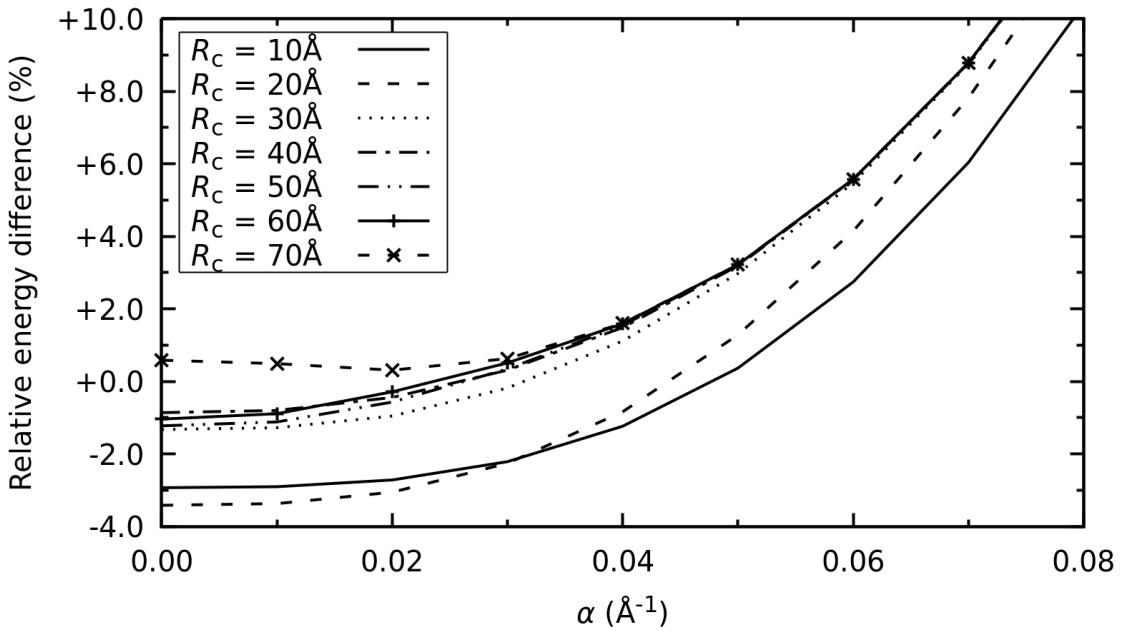


Figure 5.2: Relative difference in electrostatic energy between the Wolf method and Ewald summation for different Wolf parameters R_c as a function of α . The parameters for the Ewald summation are chosen such that a relative precision of 10^{-6} is achieved. The energy is calculated for methanol at a (typical gas) density of 2.66 kg/m^3 at 600K using the force field from [35]. The optimal value of α is in the range from 0 \AA^{-1} to 0.03 \AA^{-1} .

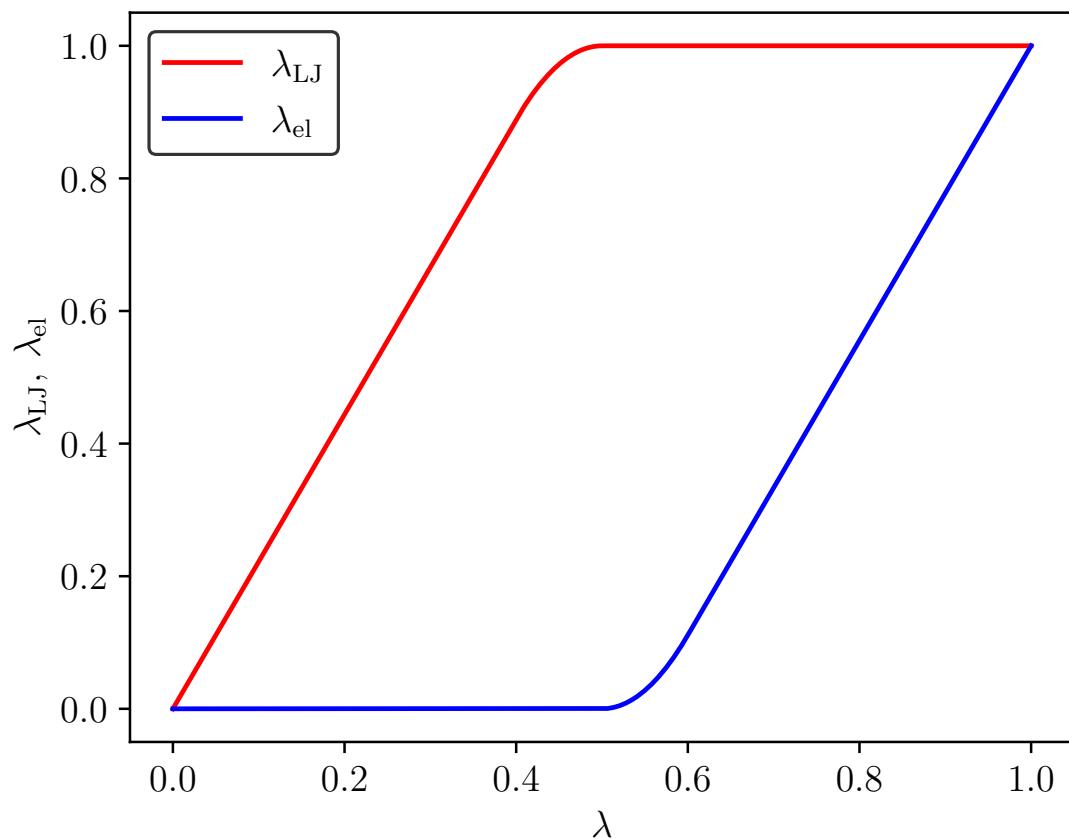


Figure 5.3: Scaling of LJ and electrostatic interactions as a function of λ for the computation of $\langle \frac{\partial U}{\partial \lambda} \rangle$ which is used for thermodynamic integration.

Chapter 6

Ensembles

6.1 Detailed balance

Detailed balance [13] implies that the flow from state o to state n :

$$\mathcal{K}(o \rightarrow n) = \mathcal{N}(o) \times \alpha(o \rightarrow n) \times \text{acc}(o \rightarrow n) \quad (6.1)$$

is equal to the flow from state n to state o :

$$\mathcal{K}(n \rightarrow o) = \mathcal{N}(n) \times \alpha(n \rightarrow o) \times \text{acc}(n \rightarrow o) \quad (6.2)$$

where $\mathcal{N}(o)$ is the probability density of finding the system in state o , $\alpha(o \rightarrow n)$ is the probability of performing a trial move from state o to state n and $\text{acc}(o \rightarrow n)$ is the probability to accept this trial move. Here, we choose α to be symmetric and in that case we have:

$$\frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} \quad (6.3)$$

which results in the Metropolis algorithm in the following acceptance rule:

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{\mathcal{N}(n)}{\mathcal{N}(o)} \right) \quad (6.4)$$

In the following sections, we will use this as our acceptance rule.

6.2 Scaled Coordinates

A typical partition function has the form:

$$Q = \frac{1}{\Lambda^{3N} N!} \int d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)] \quad (6.5)$$

where λ is the thermal wavelength, N is the number of molecules, \mathbf{r} the coordinates of the molecules, $\beta = 1/(k_B T)$, and U an energy function depending only on the coordinates of the molecules. In

a cubic system, $V = L^3$, it is convenient to use reduced coordinates $s_{x,y,z} = r_{x,y,z}/L$. And the partition function becomes:

$$Q = \frac{V^N}{\Lambda^{3N} N!} \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N; V)] \quad (6.6)$$

where U now depends on the volume of the system V because of the rescaling of \mathbf{r} . For convenience, we will omit the volume as an argument in the energy: $U(\mathbf{s}^N; V) = U(\mathbf{s}^N)$.

6.3 The *NVT* Ensemble

6.3.1 Partition Function

The partition function for the *NVT* ensemble is [13]:

$$Q_{NVT} = \frac{V^N}{\Lambda^{3N} N!} \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \quad (6.7)$$

with corresponding probability density:

$$\mathcal{N}(\mathbf{s}^N) \propto \exp[-\beta U(\mathbf{s}^N)] \quad (6.8)$$

where N is the number of molecules, V is the volume of the system, Λ is the thermal wavelength, \mathbf{s}^N is the scaled coordinate vector of all molecules, and U is the potential energy.

6.3.2 Acceptance Rules

In the *NVT* ensemble, there are three possible Monte Carlo trial moves: translation of a molecule, rotation of a molecule and torsion in a molecule. Using Eqs. (6.4) and (6.8) the acceptance rules for the *NVT* ensemble can be derived.

Translation, Rotation and Torsion Trial Moves

Performing those trial moves only result in an energy change of the system and therefore:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]) \quad (6.9)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U(\mathbf{s}_n^N) - U(\mathbf{s}_o^N). \quad (6.10)$$

6.4 The CFC*NVT* Ensemble

6.4.1 Partition Function

The partition function for the CFC*NVT* ensemble, where one fractional molecule is added to the conventional *NVT* ensemble, is [2]:

$$Q_{NVT}^{\text{CFC}} = \frac{V^{N+1}}{\Lambda^{3(N+1)} N!} \int_0^1 d\lambda \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \int d\mathbf{s}_{\text{frac}} \exp[-\beta U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.11)$$

with corresponding probability density:

$$\mathcal{N}(\mathbf{s}^N, \mathbf{s}_{\text{frac}}, \lambda) \propto \exp[-\beta(U(\mathbf{s}^N) + U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda))] \quad (6.12)$$

where N is the number of whole molecules, V is the volume of the system, Λ is the thermal wavelength, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy (excluding the interactions of the fractional molecule), U_{frac} is the potential energy of the fractional molecule(s) (this term also includes the interactions between fractional molecules in case there is more than one fractional molecule in the system), $\lambda \in [0, 1]$ is the fractional parameter, and \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule.

6.4.2 Acceptance Rules

In addition to the trial moves from the conventional NVT ensemble, there is a trial move that changes the value of the fractional parameter λ . Using Eq. (6.4) and (6.12) the acceptance rules for the CFCNVT ensemble can be derived.

Translation, Rotation and Torsion Trial Moves

Performing these trial moves only result in an energy change of the system and therefore:

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta\Delta U]) \quad (6.13)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U(\mathbf{s}_n^N) - U(\mathbf{s}_o^N) + U_{\text{frac}}(\mathbf{s}_{\text{frac},n}, \mathbf{s}_n^N, \lambda) - U_{\text{frac}}(\mathbf{s}_{\text{frac},o}, \mathbf{s}_o^N, \lambda). \quad (6.14)$$

Lambda Trial Move

Changing the value of λ only results in an energy change and we have the additional constraint that λ should be between 0 and 1. If this is the case we have the following acceptance rule (otherwise we immediately reject):

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta\Delta U]) \quad (6.15)$$

where ΔU depends only on the difference in λ because the positions of the molecules are unchanged:

$$\Delta U = U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda_n) - U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda_o). \quad (6.16)$$

6.4.3 Chemical Potential

In the CFCNVT ensemble it is possible to calculate the chemical potential from the probability distribution of the fractional parameter [2]:

$$\mu_{NVT}^{\text{CFC}} = -k_B T \ln \left\langle \frac{V/\Lambda^3}{N+1} \right\rangle - k_B T \ln \left(\frac{p(\lambda=1)}{p(\lambda=0)} \right) \quad (6.17)$$

where $p(\lambda=1)$ and $p(\lambda=0)$ are the probabilities that λ takes the values 1 or 0, respectively.

6.5 The *NPT* Ensemble

6.5.1 Partition Function

The partition function for the *NPT* ensemble is [13]:

$$Q_{NPT} = \frac{\beta P}{\Lambda^{3N} N!} \int dV V^N \exp[-\beta PV] \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \quad (6.18)$$

with corresponding probability density:

$$\mathcal{N}(\mathbf{s}^N, V) \propto V^N \exp[-\beta PV] \exp[-\beta U(\mathbf{s}^N)] \quad (6.19)$$

where N is the number of molecules, V is the volume of the system, P is the imposed pressure, Λ is the thermal wavelength, \mathbf{s}^N is the scaled coordinate vector of all molecules, and U is the potential energy.

6.5.2 Acceptance Rules

In addition to the trial moves from the conventional *NVT* ensemble, there is a trial move that changes the volume of the system. Using Eq. (6.4) and (6.19) the acceptance rules for the *NPT* ensemble can derived.

Translation, Rotation and Torsion Trial Moves

Performing these trial moves only result in an energy change of the system and therefore:

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta \Delta U]) \quad (6.20)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U(\mathbf{s}_n^N) - U(\mathbf{s}_o^N). \quad (6.21)$$

Volume Changes

For a volume change, $V_n = V_o + \Delta V$ we have:

$$\frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \frac{V_n^N \exp[-\beta PV_n] \exp[-\beta U_n(\mathbf{s}^N)]}{V_o^N \exp[-\beta PV_o] \exp[-\beta U_o(\mathbf{s}^N)]} \quad (6.22)$$

so that the acceptance rule becomes:

$$\text{acc}(o \rightarrow n) = \min \left(1, \left(\frac{V_n}{V_o} \right)^N \exp[-\beta P(V_n - V_o)] \exp[-\beta \Delta U] \right) \quad (6.23)$$

$$= \min \left(1, \exp[-\beta P(V_n - V_o) - \beta \Delta U + N \ln \left(\frac{V_n}{V_o} \right)] \right) \quad (6.24)$$

where it should be noted that ΔU only depends on the change in volume because of the rescaling of the coordinates \mathbf{r} :

$$\Delta U = U_n(\mathbf{s}^N) - U_o(\mathbf{s}^N). \quad (6.25)$$

6.5.3 Chemical Potential

Using the Widom test particle insertion method the chemical potential can be calculated [15]. By randomly placing a molecule in the simulation box, the excess chemical potential can be determined from the following ensemble average:

$$\mu_{\text{excess}} = -k_B T \ln \langle \exp[-\beta \Delta U^+] \rangle \quad (6.26)$$

where ΔU^+ is the energy of the test molecule with the rest of the system.

6.5.4 Partial Molar Enthalpy

The partial molar excess enthalpy can be obtained using a test molecule to determine the following ensemble average [16]:

$$h_{\text{excess}} = -k_B T + \frac{\langle (\Delta U^+ + U + PV) V \exp[-\beta \Delta U^+] \rangle}{V \exp[-\beta \Delta U^+]} - \langle U + PV \rangle \quad (6.27)$$

where ΔU^+ is the energy of the test molecule with the rest of the system and U the total energy of the system (excluding the test molecule).

6.5.5 Partial Molar Volume

The partial molar volume can be obtained using a test molecule to determine the following ensemble average [16]:

$$v = \frac{\langle V^2 \exp[-\beta \Delta U^+] \rangle}{\langle V \exp[-\beta \Delta U^+] \rangle} - \langle V \rangle \quad (6.28)$$

where ΔU^+ is the energy of the test molecule with the rest of the system.

6.6 The CFCNPT Ensemble

6.6.1 Partition Function

The partition function for the CFCNPT ensemble, where one fractional molecule is added to the conventional NPT ensemble, is [2]:

$$Q_{\text{NPT}}^{\text{CFC}} = \frac{\beta P}{\Lambda^{3(N+1)} N!} \int_0^1 d\lambda \int dV V^{N+1} \exp[-\beta PV] \times \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \int d\mathbf{s}_{\text{frac}} \exp[-\beta U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.29)$$

with corresponding probability density:

$$\mathcal{N}(s^N, s_{\text{frac}}, \lambda) \propto V^{N+1} \exp[-\beta PV] \exp[-\beta U(\mathbf{s}^N)] \exp[-\beta U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.30)$$

where N is the number of whole molecules, V is the volume of the system, P is the imposed pressure, Λ is the thermal wavelength, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential

energy (excluding the interactions of the fractional molecule), U_{frac} is the potential energy of the fractional molecule(s) (this term also includes the interactions between fractional molecules in case there is more than one fractional molecule in the system), $\lambda \in [0, 1]$ is the fractional parameter, and \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule.

6.6.2 Acceptance Rules

In addition to the trial moves from the conventional *NPT* ensemble, there is a trial move that changes the value of the fractional parameter λ . Using Eq. (6.4) and (6.30) the acceptance rules for the CFC*NPT* ensemble can be derived.

Translation, Rotation and Torsion Trial Moves

Performing these trial moves only result in an energy change of the system and therefore:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]) \quad (6.31)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U(\mathbf{s}_n^N) - U(\mathbf{s}_o^N) + U_{\text{frac}}(\mathbf{s}_{\text{frac},n}, \mathbf{s}_n^N, \lambda) - U_{\text{frac}}(\mathbf{s}_{\text{frac},o}, \mathbf{s}_o^N, \lambda). \quad (6.32)$$

Volume Changes

The acceptance rule for a volume change is similar to that in the conventional *NPT* ensemble with $N \rightarrow N + 1$:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \left(\frac{V_n}{V_o} \right)^{N+1} \exp[-\beta P(V_n - V_o)] \exp[-\beta \Delta U] \right) \quad (6.33)$$

$$= \min \left(1, \exp \left[-\beta P(V_n - V_o) - \beta \Delta U + (N+1) \ln \left(\frac{V_n}{V_o} \right) \right] \right) \quad (6.34)$$

where it should be noted that ΔU only depends on the change in volume because of the rescaling of the coordinates \mathbf{r} and \mathbf{r}_{frac} :

$$\Delta U = U_n(\mathbf{s}^N) - U_o(\mathbf{s}^N) + U_{\text{frac},n}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda;) - U_{\text{frac},o}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda). \quad (6.35)$$

Lambda Trial Move

Changing the value of λ only results in an energy change and we have the additional constraint that λ should be between 0 and 1. If this is the case, we have the following acceptance rule (otherwise we immediately reject):

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]) \quad (6.36)$$

where ΔU depends only on the difference in λ because the positions of the molecules are unchanged:

$$\Delta U = U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda_n) - U_{\text{frac}}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda_o). \quad (6.37)$$

6.6.3 Chemical Potential

In the CFCNPT ensemble it is possible to calculate the chemical potential from the probability distribution of the fractional parameter [2]:

$$\mu_{NPT}^{\text{CFC}} = -k_B T \ln \left\langle \frac{V/\Lambda^3}{N+1} \right\rangle - k_B T \ln \left(\frac{p(\lambda=1)}{p(\lambda=0)} \right) \quad (6.38)$$

where $p(\lambda=1)$ and $p(\lambda=0)$ are the probabilities that λ takes the values 1 or 0, respectively.

6.6.4 Excess Chemical Potential from $\partial U / \partial \lambda$

The ensemble average $\partial U / \partial \lambda$ of a species (i) can be used to calculate the excess chemical potential in the NPT or NVT ensemble using:

$$\Delta A_{NVT} = \Delta G_{NPT} = \mu_i^{\text{ex}} = \int_0^1 \left\langle \frac{\partial U}{\partial \lambda} \right\rangle d\lambda \quad (6.39)$$

Computation of this derivative in Brick-CFCMC uses the exact derivatives of the total potential energy U (Lennard-Jones, electrostatic, tail corrections, etc.) with respect to the fractional parameter λ . Note that, calculation of the average $\partial U / \partial \lambda$ can only be used in the NPT or NVT ensembles with exactly one fractional (it is allowed for this fractional to consist of multiple molecules, e.g. Na^+ and Cl^- ions). Intramolecular interactions are not scaled. In Brick-CFCMC, the derivative (integrand) will be calculated but one has to calculate the integral themselves (which can be done easily with the tool by executing `brick integrate dUdl` or by using `alchemlyb` [36]). Note that the output of Brick-CFCMC can be read in directly by `alchemlyb` [36]. This section shows the exact derivatives that are used in the calculation of $\partial U / \partial \lambda$.

Lennard-Jones

The derivative of the Lennard-Jones energy with respect to λ_m (between site i of molecule m and site j of molecule n) is equal to:

$$\begin{aligned} \frac{\partial U_{\text{LJ},ij}}{\partial \lambda_m} &= 4\epsilon_{ij}\lambda_n \left(\frac{1}{\alpha_{\text{LJ}}(1-\lambda_t)^b + (r_{ij}/\sigma_{ij})^c} \right)^{6/c} \left[\left(\frac{1}{\alpha_{\text{LJ}}(1-\lambda_t)^b + (r_{ij}/\sigma_{ij})^c} \right)^{6/c} - 1 \right. \\ &\quad \left. + \frac{6\lambda_t b \alpha_{\text{LJ}}}{c(\alpha_{\text{LJ}}(1-\lambda_t)^b + (r_{ij}/\sigma_{ij})^c)} (1-\lambda_t)^{b-1} \left(2 \left(\frac{1}{\alpha_{\text{LJ}}(1-\lambda_t)^b + (r_{ij}/\sigma_{ij})^c} \right)^{6/c} - 1 \right) \right] \end{aligned} \quad (6.40)$$

where $\lambda_t = \lambda_m \lambda_n$. Since the tail correction also depends on λ this derivative needs to be calculated as well:

$$\frac{\partial U_{\text{tail},m}}{\partial \lambda_m} = \frac{8\pi}{V} \sum_{m,n} (N_m N_{f,n} + N_n N_{f,m} + 2\lambda_n N_{f,n} N_{f,m}) \epsilon_{mn} \left[\frac{\sigma_{mn}^{12}}{9R_c^9} - \frac{\sigma_{mn}^6}{3R_c^3} \right] \quad (6.41)$$

where $N_{f,m}$ is the number of molecules of type m in the fractional.

Ewald Method

Starting from the expression for the Fourier part of the Ewald method, the derivative with respect to λ_m of (fractional) molecule m is:

$$\frac{\partial U_{\text{Fourier}}}{\partial \lambda_m} = \frac{8\pi}{V} \sum_{\mathbf{k}} \frac{1}{k^2} \exp\left[\frac{k^2}{4\alpha_{\text{el}}}\right] \left[\left(\sum_i \lambda_i q_i \cos(i\mathbf{k} \cdot \mathbf{r}_i) \right) \left(\sum_j q_j \cos(i\mathbf{k} \cdot \mathbf{r}_j) \right) + \left(\sum_i \lambda_i q_i \sin(i\mathbf{k} \cdot \mathbf{r}_i) \right) \left(\sum_j q_j \sin(i\mathbf{k} \cdot \mathbf{r}_j) \right) \right] \quad (6.42)$$

in which indices i and j run over all atoms in the system and all atoms in molecule m , respectively. This derivative is stored in such a way that when a change is applied to a single molecule, only terms related to this single molecule have to be recomputed [30]. The real space energy of the Ewald summation between sites i and j of molecules m and n is:

$$U_{\text{real},ij} = \lambda_t q_i q_j \frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t)))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} \quad (6.43)$$

where $\lambda_t = \lambda_m \lambda_n$. The derivative with respect to λ_m is then:

$$\frac{\partial U_{\text{real},ij}}{\partial \lambda_m} = \lambda_n q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t)))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} + \lambda_t (X(r_{ij})) \right] \quad (6.44)$$

with

$$X(r) = \frac{\beta_{\text{el}} \left[(r + \beta_{\text{el}}(1 - \lambda_t)) \frac{2\alpha_{\text{el}}}{\sqrt{\pi}} \exp[-\alpha_{\text{el}}^2(r + \beta_{\text{el}}(1 - \lambda_t))^2] + \text{erfc}(\alpha_{\text{el}}(r + \beta_{\text{el}}(1 - \lambda_t))) \right]}{(r + \beta_{\text{el}}(1 - \lambda_t))^2} \quad (6.45)$$

The exclusion term of the Ewald summation between atoms i and j in molecule m is obtained using:

$$U_{\text{exclusion},ij} = \lambda_t q_i q_j \frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t))) - 1}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} \quad (6.46)$$

where $\lambda_t = \lambda_m^2$. Hence the derivative with respect to λ_m is obtained as:

$$\begin{aligned} \frac{\partial U_{\text{exclusion},ij}}{\partial \lambda_m} &= 2\lambda_m q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t)))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} + \lambda_t (X(r_{ij})) \right. \\ &\quad \left. - \frac{r_{ij} + \beta_{\text{el}}(1 - \lambda_t) + \lambda_t \beta}{(r_{ij} + \beta_{\text{el}}(1 - \lambda_t))^2} \right] \end{aligned} \quad (6.47)$$

The self energy term is:

$$U_{\text{self}} = G_{\text{self}} \sum_i \lambda_i^2 q_i^2 \quad (6.48)$$

where index i runs over all atoms in the system and the values of λ_i are set to 1 for atoms of whole molecules. G_{self} is a constant prefactor of the self energy term that depends only on the chosen electrostatic method and parameters. For the Ewald method:

$$G_{\text{self}} = -\frac{\alpha}{\sqrt{\pi}} \quad (6.49)$$

And the derivative with respect to λ_m of molecule m becomes:

$$\frac{\partial U_{\text{self}}}{\partial \lambda_m} = 2\lambda_m G_{\text{self}} \sum_j q_j^2 \quad (6.50)$$

in which index j runs over all atoms in the fractional molecule m .

Wolf Method

The derivative of the short ranged real-space electrostatic energy between site i of molecule m and site j of molecule n is computed as:

$$\begin{aligned} \frac{\partial U_{\text{real},ij}}{\partial \lambda_m} = & \lambda_n q_i q_j \left[\frac{\operatorname{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t)))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} \right. \\ & - \frac{\operatorname{erfc}(\alpha_{\text{el}}(R_c + \beta_{\text{el}}(1 - \lambda_t)))}{R_c + \beta_{\text{el}}(1 - \lambda_t)} \\ & \left. + \lambda_t (X(r_{ij}) - X(R_c)) \right] \end{aligned} \quad (6.51)$$

and X is equal to 6.45. The exclusion term in the Wolf method between sites i and j of the fractional molecule m is computed as:

$$U_{\text{exclusion},ij} = \lambda_t q_i q_j \left[\frac{\operatorname{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t))) - 1}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} - \frac{\operatorname{erfc}(\alpha_{\text{el}}(R_c + \beta_{\text{el}}(1 - \lambda_t)))}{R_c + \beta_{\text{el}}(1 - \lambda_t)} \right] \quad (6.52)$$

in which $\lambda_t = \lambda_m^2$, and the derivative with respect to λ_m is calculated using:

$$\begin{aligned} \frac{\partial U_{\text{exclusion},ij}}{\partial \lambda_m} = & 2\lambda_m q_i q_j \left[\frac{\operatorname{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t)))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} - \frac{\operatorname{erfc}(\alpha_{\text{el}}(R_c + \beta_{\text{el}}(1 - \lambda_t)))}{R_c + \beta_{\text{el}}(1 - \lambda_t)} \right. \\ & \left. + \lambda_t (X(r_{ij}) - X(R_c)) - \frac{r_{ij} + \beta_{\text{el}}(1 - \lambda_t) + \lambda_t \beta}{(r_{ij} + \beta_{\text{el}}(1 - \lambda_t))^2} \right] \end{aligned} \quad (6.53)$$

Finally, the derivative from the selfterm is equal to the expression in 6.48 with:

$$G_{\text{self}} = -\left(\frac{\operatorname{erfc}(\alpha R_c)}{2R_c} + \frac{\alpha}{\sqrt{\pi}} \right) \quad (6.54)$$

Fennell-Gezelter (Damped Shifted Force Potential) Method

For the DSF potential the same derivatives hold as for the Wolf method, except for the real term. The derivative of the real term with respect to λ_m is equal to:

$$\frac{\partial U_{\text{real},ij}}{\partial \lambda_m} = \lambda_n q_i q_j \left[\frac{\operatorname{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_t)))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_t)} - \frac{\operatorname{erfc}(\alpha_{\text{el}}(R_c + \beta_{\text{el}}(1 - \lambda_t)))}{R_c + \beta_{\text{el}}(1 - \lambda_t)} + \left(\frac{\operatorname{erfc}(\alpha_{\text{el}} R_c)}{R_c^2} + \frac{2\alpha_{\text{el}} \exp[-\alpha_{\text{el}}^2 R_c^2]}{\sqrt{\pi}} \right) (r_{ij} - R_c) + \lambda_t (X(r_{ij}) - X(R_c)) \right] \quad (6.55)$$

and X is equal to the expression in 6.45.

6.6.5 Partial Molar Enthalpy

The expression to obtain the partial molar enthalpy in the CFCNPT is derived in the Supporting Information of Ref. [2]. We state the expression here:

$$h_{NPT}^{\text{CFC}} = -k_B T + \langle H(\lambda = 1) \rangle - \frac{\langle \frac{H}{V}(\lambda = 0) \rangle}{\langle \frac{1}{V}(\lambda = 0) \rangle} \quad (6.56)$$

where $H(\lambda = 1)$ is the total enthalpy in the system when $\lambda = 1$, $\frac{H}{V}(\lambda = 0)$ the total enthalpy divided by the volume when $\lambda = 0$, and $\frac{1}{V}(\lambda = 0)$ is the inverse volume when $\lambda = 0$.

6.6.6 Partial Molar Volume

The expression to obtain the partial molar volume in the CFCNPT is derived in the Supporting Information of Ref. [2]. We state the expression here:

$$v_{NPT}^{\text{CFC}} = \left\langle V(\lambda = 1) \right\rangle - \left\langle \frac{1}{V}(\lambda = 0) \right\rangle^{-1} \quad (6.57)$$

where $V(\lambda = 1)$ is the volume of the system when $\lambda = 1$, and $\frac{1}{V}(\lambda = 0)$ the inverse volume when $\lambda = 0$.

6.7 The CFC Gibbs Ensembles

Here, we consider two versions of the Gibbs ensemble [37] with the CFC method [38]: the CFC Gibbs Ensemble at constant total volume and the CFC Gibbs Ensemble at constant pressure. For pure component systems, one has to use the constant volume version (by specifying in the input a value for the pressure that is negative).

6.7.1 Partition Function CFC Gibbs Ensemble at Constant Total Volume

The partition function for the CFC Gibbs Ensemble at constant total volume is [38]:

$$\begin{aligned}
Q_{\text{GE}}^{\text{CFC}} = & \frac{1}{\Lambda^{3(N_T+1)} N_T!} \sum_{i=1}^2 \sum_{N_1=0}^{N_T} \frac{N_T!}{N_1!(N_T - N_1)!} \int_0^1 d\lambda \\
& \times \int_0^{V_T} dV_1 V_1^{N_1 + \delta_{i,1}} (V_T - V_1)^{N_T - N_1 + \delta_{i,2}} \\
& \times \int d\mathbf{s}^{N_1} \exp[-\beta U_1(\mathbf{s}^{N_1})] \\
& \times \int d\mathbf{s}^{N_T - N_1} \exp[-\beta U_2(\mathbf{s}^{N_T - N_1})] \\
& \times \left(\delta_{i,1} \int d\mathbf{s}_{\text{frac}} \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_1}, \lambda)] \right. \\
& \left. + \delta_{i,2} \int d\mathbf{s}_{\text{frac}} \exp[-\beta U_{\text{frac},2}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_T - N_1}, \lambda)] \right)
\end{aligned} \tag{6.58}$$

with corresponding probability density:

$$\begin{aligned}
& \mathcal{N}(N_1, N_2, \mathbf{s}^{N_1}, \mathbf{s}^{N_2}, \mathbf{s}_{\text{frac}}^i, V_1, V_2, \lambda, i) \propto \\
& \frac{V_1^{N_1 + \delta_{i,1}} V_2^{N_2 + \delta_{i,2}}}{N_1! N_2!} \exp[-\beta U_1(\mathbf{s}^{N_1})] \exp[-\beta U_2(\mathbf{s}^{N_2})] \\
& \times \left(\delta_{i,1} \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_1}, \lambda)] + \delta_{i,2} \exp[-\beta U_{\text{frac},2}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_2}, \lambda)] \right)
\end{aligned} \tag{6.59}$$

where N_j and V_j are the number of whole molecules and volume of simulation box j , N_T is the total number of whole molecules in the two simulation boxes, V_T is the total volume of the two simulation boxes, Λ is the thermal wavelength, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy, U_{frac} is the potential energy of the fractional molecule, $\lambda \in [0, 1]$ is the fractional parameter, \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule, and the δ -function is used to indicate if the fractional molecule is in simulation box i ($\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ if $i = j$).

6.7.2 Partition Function CFC Gibbs Ensemble at Constant Pressure

The partition function for the CFC Gibbs Ensemble at constant pressure is [38]:

$$\begin{aligned}
Q_{\text{GE}}^{\text{CFC}} = & \frac{\beta P}{\Lambda^{3(N_T+1)} N_T!} \sum_{i=1}^2 \sum_{N_1=0}^{N_T} \frac{N_T!}{N_1!(N_T-N_1)!} \int_0^1 d\lambda \\
& \times \int dV_1 V_1^{N_1+\delta_{i,1}} \exp[-\beta PV_1] \\
& \times \int dV_2 V_2^{N_T-N_1+\delta_{i,2}} \exp[-\beta PV_2] \\
& \times \int d\mathbf{s}^{N_1} \exp[-\beta U_1(\mathbf{s}^{N_1})] \\
& \times \int d\mathbf{s}^{N_T-N_1} \exp[-\beta U_2(\mathbf{s}^{N_T-N_1})] \\
& \times \left(\delta_{i,1} \int d\mathbf{s}_{\text{frac}} \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_1}, \lambda)] \right. \\
& \quad \left. + \delta_{i,2} \int d\mathbf{s}_{\text{frac}} \exp[-\beta U_{\text{frac},2}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_T-N_1}, \lambda)] \right)
\end{aligned} \tag{6.60}$$

with corresponding probability density:

$$\begin{aligned}
\mathcal{N}(N_1, N_2, \mathbf{s}^{N_1}, \mathbf{s}^{N_2}, s_{\text{frac},i}, V_1, V_2, \lambda, i) \propto & \frac{V_1^{N_1+\delta_{i,1}} V_2^{N_2+\delta_{i,2}}}{N_1! N_2!} \\
& \times \exp[-\beta PV_1] \exp[-\beta PV_2] \exp[-\beta U_1(\mathbf{s}^{N_1})] \exp[-\beta U_2(\mathbf{s}^{N_2})] \\
& \times \left(\delta_{i,1} \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_1}, \lambda)] + \delta_{i,2} \exp[-\beta U_{\text{frac},2}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_2}, \lambda)] \right)
\end{aligned} \tag{6.61}$$

where N_j and V_j are the number of whole molecules and volume of simulation box j , N_T is the total number of whole molecules in the two simulation boxes, P is the imposed pressure, Λ is the thermal wavelength, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy, U_{frac} is the potential energy of the fractional molecule, $\lambda \in [0, 1]$ is the fractional parameter, \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule, and the δ -function is used to indicate if the fractional molecule is in simulation box i ($\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ if $i = j$).

6.7.3 Acceptance Rules

In the CFC Gibbs Ensembles we consider the following trial moves: translation of a molecule, rotation of a molecule, torsion in a molecule, volume changes, changing the value of the fractional parameter λ and molecule transfer trial moves. Using Eq. (6.4) and the probability densities the acceptance rules can be derived.

Translation, Rotation and Torsion Trial Moves

Performing these trial moves only result in an energy change of the system in the box that the selected molecule is in. The acceptance rule for performing one of those trial moves on a molecule

in box i while the fractional molecule is in box j is than:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U]) \quad (6.62)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U_i(\mathbf{s}_n^N) - U_i(\mathbf{s}_o^N) + \delta_{i,j} \Delta U_{\text{frac}} \quad (6.63)$$

$$\Delta U_{\text{frac}} = U_{\text{frac},j}(\mathbf{s}_{\text{frac},n}, \mathbf{s}_n^{N_j}, \lambda) - U_{\text{frac},j}(\mathbf{s}_{\text{frac},o}, \mathbf{s}_o^{N_j}, \lambda) \quad (6.64)$$

Volume Changes at Constant Total Volume

If the total volume V_T is constant, we always have: $V_{1,n} = V_{1,o} + \Delta V \Leftrightarrow V_{2,n} = V_{2,o} - \Delta V$. We derive the acceptance rule for the case when the fractional molecule is in box 1:

$$\frac{\mathcal{N}(\text{n})}{\mathcal{N}(\text{o})} = \frac{\frac{V_{1,n}^{N_1+1} V_{2,n}^{N_2}}{N_1! N_2!} \exp[-\beta U_1(\mathbf{s}^{N_1})] \exp[-\beta U_2(\mathbf{s}^{N_2})] \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_1}, \lambda)]}{\frac{V_{1,o}^{N_1+1} V_{2,o}^{N_2}}{N_1! N_2!} \exp[-\beta U_1(\mathbf{s}^{N_1})] \exp[-\beta U_2(\mathbf{s}^{N_2})] \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_1}, \lambda)]} \quad (6.65)$$

$$= \frac{V_{1,n}^{N_1+1} V_{2,n}^{N_2}}{V_{1,o}^{N_1+1} V_{2,o}^{N_2}} \exp[-\beta \Delta U] \quad (6.66)$$

which can be generalized to obtain the acceptance rule for a volume change when the fractional molecule is in box i :

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{V_{1,n}^{N_1+\delta_{i,1}} V_{2,n}^{N_2+\delta_{i,2}}}{V_{1,o}^{N_1+\delta_{i,1}} V_{2,o}^{N_2+\delta_{i,2}}} \exp[-\beta \Delta U] \right) \quad (6.67)$$

where it should be noted that ΔU depends only on the change in volume because of the rescaling of the coordinates \mathbf{r} and \mathbf{r}_{frac} :

$$\Delta U = U_{1,n}(\mathbf{s}^{N_1}) + U_{2,n}(\mathbf{s}^{N_2}) - U_{1,o}(\mathbf{s}^{N_1}) - U_{2,o}(\mathbf{s}^{N_2}) \quad (6.68)$$

$$+ U_{\text{frac},i,n}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_i}, \lambda) - U_{\text{frac},i,o}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_i}, \lambda). \quad (6.69)$$

Volume Changes at Constant Pressure

If the pressure is constant, a volume change can be done to either box independently. Therefore, the acceptance rule is very similar to that for the CFCNPT ensemble. For changing the volume of box i ($V_{i,n} = V_{i,o} + \Delta V$) while the fractional molecule is in box j the acceptance rule is:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \left(\frac{V_{i,n}}{V_{i,o}} \right)^{N+\delta_{i,j}} \exp[-\beta P(V_{i,n} - V_{i,o})] \exp[-\beta \Delta U] \right) \quad (6.70)$$

$$= \min \left(1, \exp[-\beta P(V_{i,n} - V_{i,o}) - \beta \Delta U + (N + \delta_{i,j}) \ln \left(\frac{V_{i,n}}{V_{i,o}} \right)] \right) \quad (6.71)$$

where it should be noted that ΔU depends only on the change in volume because of the rescaling of the coordinates \mathbf{r} and \mathbf{r}_{frac} :

$$\Delta U = U_{i,n}(\mathbf{s}^{N_i}) - U_{i,o}(\mathbf{s}^{N_i}) + \delta_{i,j} \Delta U_{\text{frac}} \quad (6.72)$$

$$\Delta U_{\text{frac}} = U_{\text{frac},j,n}(\mathbf{s}_{\text{frac}}^j, \mathbf{s}^{N_j}, \lambda) - U_{\text{frac},j,o}(\mathbf{s}_{\text{frac}}^j, \mathbf{s}^{N_j}, \lambda) \quad (6.73)$$

Lambda Trial Move

Changing the value of λ only results in an energy change and we have the additional constraint that λ should be between 0 and 1. If this is the case, we have the following acceptance rule (otherwise we reject immediately):

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta \Delta U]) \quad (6.74)$$

when the fractional molecule is in box i and ΔU depends only on the difference in λ because the positions of the molecules are unchanged:

$$\Delta U = U_{\text{frac},i}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda_n) - U_{\text{frac},i}(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda_o) \quad (6.75)$$

Transferring (Swapping) a Fractional Group

Here we derive the acceptance rule for transferring/swapping a Fractional Group from box 1 to box 2 while keeping λ fixed. We then have:

$$\frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \frac{\frac{V_1^{N_1} V_2^{N_2+1}}{N_1! N_2!} \exp[-\beta U_1(\mathbf{s}^{N_1})] \exp[-\beta U_2(\mathbf{s}^{N_2})] \exp[-\beta U_{\text{frac},2}(\mathbf{s}_{\text{frac},n}, \mathbf{s}^{N_2}, \lambda)]}{\frac{V_1^{N_1+1} V_2^{N_2}}{N_1! N_2!} \exp[-\beta U_1(\mathbf{s}^{N_1})] \exp[-\beta U_2(\mathbf{s}^{N_2})] \exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac},o}, \mathbf{s}^{N_1}, \lambda)]} \quad (6.76)$$

$$= \frac{V_2}{V_1} \exp[-\beta (U_{\text{frac},2,n}(\mathbf{s}_{\text{frac}}, \mathbf{s}^{N_2}, \lambda) - U_{\text{frac},1}(\mathbf{s}_{\text{frac},o}, \mathbf{s}^{N_1}, \lambda))] \quad (6.77)$$

And this can easily be generalized for transferring the fractional molecule from box i to box $j \neq i$:

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{V_j}{V_i} \exp[-\beta \Delta U] \right) \quad (6.78)$$

where $\mathbf{s}_{\text{frac},n}$ is a random position:

$$\Delta U = U_{\text{frac},j}(\mathbf{s}_{\text{frac},n}, \mathbf{s}^{N_j}, \lambda) - U_{\text{frac},i}(\mathbf{s}_{\text{frac},o}, \mathbf{s}^{N_i}, \lambda) \quad (6.79)$$

Changing the Identity of a Fractional Group

Here we derive the acceptance rule for changing molecules in a Fractional Group in box 1 to whole molecules and changing molecules in box 2 to Fractional Molecules with λ unchanged. We then

have:

$$\frac{\mathcal{N}(\text{n})}{\mathcal{N}(\text{o})} = \frac{\frac{V_1^{N_1} V_2^{N_2}}{(N_1+1)!(N_2-1)!} \exp[-\beta U_1(\mathbf{s}_n^{N_1+1})] \exp[-\beta U_2(\mathbf{s}_n^{N_2-1})]}{\frac{V_1^{N_1} V_2^{N_2}}{N_1! N_2!} \exp[-\beta U_1(\mathbf{s}_o^{N_1})] \exp[-\beta U_2(\mathbf{s}_o^{N_2})]} \dots \frac{\exp[-\beta U_{\text{frac},2}(\mathbf{s}_{\text{frac},n}, \mathbf{s}_n^{N_2-1}, \lambda)]}{\exp[-\beta U_{\text{frac},1}(\mathbf{s}_{\text{frac},o}, \mathbf{s}_o^{N_1}, \lambda)]} \quad (6.80)$$

$$= \frac{N_2}{N_1 + 1} \exp \left[-\beta (U_1(\mathbf{s}_n^{N_1+1}) - U_1(\mathbf{s}_o^{N_1}) + U_2(\mathbf{s}_n^{N_2-1}) - U_2(\mathbf{s}_o^{N_2}) + U_{\text{frac},2}(\mathbf{s}_{\text{frac},n}, \mathbf{s}_n^{N_2-1}, \lambda) - U_{\text{frac},1}(\mathbf{s}_{\text{frac},o}, \mathbf{s}_o^{N_1}, \lambda)) \right] \quad (6.81)$$

And this can be easily generalized for changing the fractional molecule in box i to a whole and changing a molecule in box $j \neq i$ to a fractional:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{N_j}{N_i + 1} \exp[-\beta \Delta U] \right) \quad (6.82)$$

where the difference in energy is:

$$\begin{aligned} \Delta U = & U_i(\mathbf{s}_n^{N_i+1}) + U_j(\mathbf{s}_n^{N_j-1}) - U_i(\mathbf{s}_o^{N_i}) - U_j(\mathbf{s}_o^{N_j}) \\ & + U_{\text{frac},j}(\mathbf{s}_{\text{frac},n}^j, \mathbf{s}_n^{N_j-1}, \lambda) - U_{\text{frac},i}(\mathbf{s}_{\text{frac},o}^i, \mathbf{s}_o^{N_i}, \lambda). \end{aligned} \quad (6.83)$$

6.7.4 Chemical Potential

In the CFC Gibbs Ensembles it is possible to calculate the chemical potentials μ in the simulation boxes. Here, we derive the expression for the CFC Gibbs Ensemble at constant pressure. The derivation for the CFC Gibbs Ensemble at constant total volume is similar and yields the same result [38]:

$$\mu_{\text{GE},i}^{\text{CFC}} = -k_B T \ln \left\langle \frac{V_i/\Lambda^3}{N_i + 1} \right\rangle - k_B T \ln \left(\frac{p_i(\lambda = 1)}{p_i(\lambda = 0)} \right) \quad (6.84)$$

where p_i denotes the probability distribution function of λ in box i .

6.8 The CFC Reaction Ensembles

Here, we consider two versions of the Reaction Ensemble [39, 40] with the CFC method [1]: one at constant volume and one at constant pressure.

6.8.1 Partition Function CFC Reaction Ensemble at Constant Volume

The partition function for the CFC Reaction Ensemble at constant volume is [1]:

$$\begin{aligned}
Q_{\text{RE}}^{\text{CFC}} = & \sum_{N_1=0}^{\infty} \cdots \sum_{N_S=0}^{\infty} \sum_{\delta=0}^1 \int_0^1 d\lambda \\
& \times \exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta \nu_i) + (N_i + \delta \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\
& \times \exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1-\delta) \nu_i) + (N_i + (1-\delta) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\
& \times \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \int d\mathbf{s}_{\text{frac}}^{N_{\text{frac}}} \exp[-\beta U(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta)]
\end{aligned} \tag{6.85}$$

with corresponding probability density:

$$\begin{aligned}
\mathcal{N}(N, N_{\text{frac}}, \mathbf{s}^N, \mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \lambda, \delta) \propto & \\
& \exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta \nu_i) + (N_i + \delta \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\
& \times \exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1-\delta) \nu_i) + (N_i + (1-\delta) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\
& \times \exp[-\beta U(\mathbf{s}^N)] \exp[-\beta U(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta)]
\end{aligned} \tag{6.86}$$

where N_i is the number of whole molecules of component i , Λ_i is the thermal wavelength, q_i is the partition function of the isolated molecule excluding the translational part (see the Appendix of this document for details), μ_i is the chemical potential, ν_i is the stoichiometric coefficient, δ indicates the reaction step ($\delta = 0$ indicates that the fractional molecules are reactants and $\delta = 1$ indicates that the fractional molecules are products), \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy (excluding the interactions of fractional molecules), U_{frac} is the potential energy of the fractional molecule (including interactions between fractional molecules), λ is the fractional parameter, and \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule. The components are labeled such that components 1 to R are reactants, and components $R + 1$ to S are products. Multiple reactions can be defined in the Reaction Ensemble.

6.8.2 Partition Function CFC Reaction Ensemble at Constant Pressure

The partition function for the CFC Reaction Ensemble at constant pressure is [1]:

$$Q_{\text{RE}}^{\text{CFC}} = \beta P \sum_{N_1=0}^{\infty} \cdots \sum_{N_S=0}^{\infty} \sum_{\delta=0}^1 \int_0^1 d\lambda \int dV \exp[-\beta PV] \\ \times \exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta \nu_i) + (N_i + \delta \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\ \times \exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1-\delta) \nu_i) + (N_i + (1-\delta) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\ \times \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \int d\mathbf{s}_{\text{frac}}^{N_{\text{frac}}} \exp[-\beta U(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta)] \quad (6.87)$$

with corresponding probability density:

$$\mathcal{N}(N, N_{\text{frac}}, \mathbf{s}^N, \mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, V, \lambda, \delta) \propto \\ V \exp[-\beta PV] \exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta \nu_i) + (N_i + \delta \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\ \times \exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1-\delta) \nu_i) + (N_i + (1-\delta) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right] \\ \times \exp[-\beta U(\mathbf{s}^N)] \exp[-\beta U(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta)] \quad (6.88)$$

where N_i is the number of whole molecules of component i , Λ_i is the thermal wavelength, q_i is the partition function of the isolated molecule excluding the translational part (see the Appendix of this document for details), μ_i is the chemical potential, ν_i is the stoichiometric coefficient, δ indicates the reaction step ($\delta = 0$ indicates that the fractional molecules are reactants and $\delta = 1$ indicates the fractional molecules are products), P is the imposed pressure, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy (excluding the interactions of fractional molecules), U_{frac} is the potential energy of the fractional molecule (including interactions between fractional molecules), λ is the fractional parameter, and \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule. The components are labeled such that components 1 to R are reactants, and components $R+1$ to S are products. Multiple reactions can be defined in the Reaction Ensemble.

6.8.3 Acceptance Rules

In addition to the trial moves from the CFCNVT and CFCNPT ensembles, there are additional trial moves that involve reactions: a reaction for fractional molecules, and a reaction for whole molecules.

Translation, Rotation and Torsion Trial Moves

Performing these trial moves only result in an energy change of the system and therefore:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]) \quad (6.89)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U(\mathbf{s}_n^N) - U(\mathbf{s}_o^N) + U_{\text{frac}}(\mathbf{s}_{\text{frac},n}^{N_{\text{frac}}}, \mathbf{s}_n^N, \lambda, \delta) - U_{\text{frac}}(\mathbf{s}_{\text{frac},o}^{N_{\text{frac}}}, \mathbf{s}_o^N, \lambda, \delta). \quad (6.90)$$

Volume Changes

For a volume change the acceptance rule is similar to that in the conventional *NPT* ensemble with $N \rightarrow N + N_{\text{frac}}$:

$$\text{acc}(o \rightarrow n) = \min \left(1, \left(\frac{V_n}{V_o} \right)^{N+N_{\text{frac}}} \exp[-\beta P(V_n - V_o)] \exp[-\beta \Delta U] \right) \quad (6.91)$$

$$= \min \left(1, \exp[-\beta P(V_n - V_o) - \beta \Delta U + (N + N_{\text{frac}}) \ln \left(\frac{V_n}{V_o} \right)] \right) \quad (6.92)$$

where it should be noted that ΔU only depends on the change in volume because of the rescaling of the coordinates \mathbf{r} and \mathbf{r}_{frac} :

$$\Delta U = U_n(\mathbf{s}^N) - U_o(\mathbf{s}^N) + U_{\text{frac},n}(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta) - U_{\text{frac},o}(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta). \quad (6.93)$$

Lambda Trial Move

Changing the value of λ only results in an energy change and we have the additional constraint that λ should be between 0 and 1. If this is the case, we have the following acceptance rule (otherwise we immediately reject):

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta \Delta U]) \quad (6.94)$$

where ΔU depends only on the difference in λ because the positions of the molecules are unchanged:

$$\Delta U = U_{\text{frac}}(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda_n, \delta) - U_{\text{frac}}(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda_o, \delta) \quad (6.95)$$

Reaction for Fractional Group

In this trial move the Fractional Group of the reactant (or product) is deleted from the system and a Fractional Group of the product (or reactant) is randomly inserted. We then have:

$$\begin{aligned} \frac{\mathcal{N}(n)}{\mathcal{N}(o)} &= \frac{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta_n \nu_i) + (N_i + \delta_n \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]}{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta_o \nu_i) + (N_i + \delta_o \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1 - \delta_n) \nu_i) + (N_i + (1 - \delta_n) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]}{\exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1 - \delta_o) \nu_i) + (N_i + (1 - \delta_o) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp[-\beta U(\mathbf{s}_{\text{frac},n}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_n)]}{\exp[-\beta U(\mathbf{s}_{\text{frac},o}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_o)]} \end{aligned} \quad (6.96)$$

If we now consider the case where we have fractional reactants and change them into products ($\delta_o = 1$ and $\delta_n = o$) we have (with some re-arranging):

$$\begin{aligned} \frac{\mathcal{N}(n)}{\mathcal{N}(o)} &= \frac{\exp \left[\sum_{i=1}^R \beta \mu_i N_i + N_i \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]}{\exp \left[\sum_{i=R+1}^S \beta \mu_i N_i + N_i \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + \nu_i) + (N_i + \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]}{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \nu_i) + (N_i + \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp[-\beta U(s_{frac,n}^{N_{frac}}, s^N, \lambda, \delta_n)]}{\exp[-\beta U(s_{frac,o}^{N_{frac}}, s^N, \lambda, \delta_o)]} \\ &= \exp \left[\sum_{i=R+1}^S \beta \mu_i \nu_i + \nu_i \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \sum_{i=1}^R \beta \mu_i \nu_i + \nu_i \ln \left(\frac{V q_i}{\Lambda_i^3} \right) \right] \exp[-\beta \Delta U] \end{aligned} \quad (6.97)$$

In a chemical reaction equilibrium we have:

$$\sum_{i=1}^R \mu_i \nu_i = \sum_{i=R+1}^S \mu_i \nu_i \quad (6.98)$$

Using this, and the cancellation of the exponential and logarithm functions, we obtain the acceptance rule for changing from fractional reactants to fractional products:

$$acc(o \rightarrow n) = \min \left(1, \prod_{i=1}^R \left(\frac{V q_i}{\Lambda_i^3} \right)^{-\nu_i} \prod_{i=R+1}^S \left(\frac{V q_i}{\Lambda_i^3} \right)^{\nu_i} \exp[-\beta \Delta U] \right) \quad (6.99)$$

where the energy difference is:

$$\Delta U = U(s_{frac,n}^{N_{frac}}, s^N, \lambda, \delta_n) - U(s_{frac,o}^{N_{frac}}, s^N, \lambda, \delta_o) \quad (6.100)$$

For the reverse case, fractional products to fractional reactants, the only difference is the change in signs in front of the stoichiometric coefficients ν_i .

Reaction for Whole Molecules

In this trial move the fractional molecules of the reactant (or product) are changed to whole molecules and random fractional molecules of the product (or reactant) are changed to fractionals.

Let us consider the case where we have fractional reactants in the old configuration, we then have:

$$\begin{aligned} \frac{\mathcal{N}(n)}{\mathcal{N}(o)} &= \frac{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \nu_i + \delta_n \nu_i) + (N_i + \nu_i + \delta_n \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln(N_i + \nu_i)! \right]}{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \delta_o \nu_i) + (N_i + \delta_o \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i - \nu_i + (1 - \delta_n) \nu_i) \right]}{\exp \left[\sum_{i=R+1}^S \beta \mu_i (N_i + (1 - \delta_o) \nu_i) \right]} \dots \\ &\dots \frac{\exp \left[(N_i - \nu_i + (1 - \delta_n) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln(N_i - \nu_i)! \right]}{\exp \left[(N_i + (1 - \delta_o) \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp[-\beta U(\mathbf{s}_{\text{frac},n}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_n)]}{\exp[-\beta U(\mathbf{s}_{\text{frac},o}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_o)]} \end{aligned} \quad (6.101)$$

with $\delta_o = 1$ and $\delta_n = o$ this becomes:

$$\begin{aligned} \frac{\mathcal{N}(n)}{\mathcal{N}(o)} &= \frac{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \nu_i) + (N_i + \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln(N_i + \nu_i)! \right]}{\exp \left[\sum_{i=1}^R \beta \mu_i (N_i + \nu_i) + (N_i + \nu_i) \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp \left[\sum_{i=R+1}^S \beta \mu_i N_i + N_i \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln(N_i - \nu_i)! \right]}{\exp \left[\sum_{i=R+1}^S \beta \mu_i N_i + N_i \ln \left(\frac{V q_i}{\Lambda_i^3} \right) - \ln N_i! \right]} \\ &\times \frac{\exp[-\beta U(\mathbf{s}_{\text{frac},n}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_n)]}{\exp[-\beta U(\mathbf{s}_{\text{frac},o}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_o)]} \end{aligned} \quad (6.102)$$

Again, using the cancellation of the exponential and logarithm functions we obtain the acceptance rule for changing fractional reactants into whole molecules and whole product molecules into fractional products:

$$\text{acc}(o \rightarrow n) = \min \left(1, \prod_{i=1}^R \frac{N_i!}{(N_i + \nu_i)!} \prod_{i=R+1}^S \frac{N_i}{(N_i - \nu_i)!} \exp[-\beta \Delta U] \right) \quad (6.103)$$

where the energy difference is:

$$\Delta U = U(\mathbf{s}_{\text{frac},n}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_n) - U(\mathbf{s}_{\text{frac},o}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta_o) \quad (6.104)$$

For the reverse case, changing fractional products into whole molecules and whole reactant molecules into fractional reactants, the only difference is the change in signs in front of the stoichiometric coefficients ν_i .

6.8.4 Chemical Potential

The expression to obtain the chemical potentials in the CFC Reaction Ensembles is derived in the Supporting Information of Ref. [1]. We state the expression here:

$$\sum_{i=1}^R \nu_i \mu_i = -k_B T \ln \left\langle \prod_{i=1}^R \left(\frac{q_i}{\Lambda_i^3 \rho_i} \right)^{\nu_i} \right\rangle - k_B T \ln \left(\frac{p_R(\lambda \uparrow 1)}{p_R(\lambda \downarrow 0)} \right) \quad (6.105)$$

$$\sum_{i=R+1}^S \nu_i \mu_i = -k_B T \ln \left\langle \prod_{i=R+1}^S \left(\frac{q_i}{\Lambda_i^3 \rho_i} \right)^{\nu_i} \right\rangle - k_B T \ln \left(\frac{p_P(\lambda \uparrow 1)}{p_P(\lambda \downarrow 0)} \right) \quad (6.106)$$

where p_R and p_P are the probability distributions of λ for fractional reactants and fractional products, respectively.

If fractionals are inserted one by one, with increasing λ , it is possible to compute the chemical potentials of each component individually. For example, if the component at $\lambda = \lambda_a$ has no interaction and at $\lambda = \lambda_b$ it has full interaction, we can calculate the chemical potential:

$$\nu_i \mu_i = -k_B T \ln \left(\frac{q_i}{\Lambda_i^3 \rho_i} \right)^{\nu_i} - k_B T \ln \left(\frac{p_{R/P}(\lambda \uparrow \lambda_b)}{p_{R/P}(\lambda \downarrow \lambda_a)} \right) \quad (6.107)$$

where the probability distribution $p_{R/P}$ should correspond to whether the selected component is a reactant or product.

6.8.5 Fugacity coefficient

The expression to obtain the fugacity coefficients in the CFC Reaction Ensembles is derived in the Supporting Information of Ref. [1]. We state the expression here:

$$\prod_{i=1}^R \varphi_i^{-\nu_i} = \left(\frac{\beta y_i P}{\rho_i} \right)^{\nu_i} \frac{p_R(\lambda \uparrow 1)}{p_R(\lambda \downarrow 0)} \quad (6.108)$$

$$\prod_{i=R+1}^S \varphi_i^{-\nu_i} = \left(\frac{\beta y_i P}{\rho_i} \right)^{\nu_i} \frac{p_P(\lambda \uparrow 1)}{p_P(\lambda \downarrow 0)} \quad (6.109)$$

where p_R and p_P are the probability distributions of λ for fractional reactants and fractional products respectively, and y_i is the mole fraction of component i .

Similar to the chemical potential, if fractionals are inserted one by one (from $\lambda = \lambda_a$ to $\lambda = \lambda_b$), it is possible to calculate the fugacity coefficient of a single component:

$$\varphi_i^{-\nu_i} = \left(\frac{\beta y_i P}{\rho_i} \right)^{\nu_i} \frac{p_{R/P}(\lambda \uparrow \lambda_b)}{p_{R/P}(\lambda \downarrow \lambda_a)} \quad (6.110)$$

where the probability distribution $p_{R/P}$ should correspond to whether the selected component is a reactant or product.

6.9 The CFC Grand-Canonical and Osmotic Ensembles

Here, we consider two versions of the Grand-Canonical Ensemble [41] with the CFC method [42]: one at constant volume and one at constant pressure. The latter is often referred to as the Osmotic Ensemble.

6.9.1 Partition Function CFC Grand-Canonical Ensemble

The partition function for the CFC Grand-Canonical Ensemble, is [42]:

$$Q_{GC}^{\text{CFC}} = \sum_{N=0}^{\infty} \frac{\exp[\beta\mu N] V^{N+1}}{\Lambda^{3(N+1)} N!} \times \int_0^1 d\lambda \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \\ \times \int d\mathbf{s}_{\text{frac}} \exp[-\beta U(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.111)$$

with corresponding probability density:

$$\mathcal{N}(N, \mathbf{s}^N, s_{\text{frac}}, \lambda) \propto \frac{\exp[\beta\mu N] V^{N+1}}{\Lambda^{3(N+1)} N!} \exp[-\beta U(\mathbf{s}^N)] \exp[-\beta U(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.112)$$

where N is the number of whole molecules, V is the volume of the system, Λ is the thermal wavelength, μ is the imposed chemical potential, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy (excluding the interactions of the fractional molecules), U_{frac} is the potential energy of the fractional molecule (including interactions between fractional molecules), $\lambda \in [0, 1]$ is the fractional parameter, and \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule.

CFC Osmotic Ensemble

The partition function for the Osmotic Ensemble, is [42]:

$$Q_{GC}^{\text{CFC}} = \sum_{N=0}^{\infty} \frac{\exp[\beta\mu N] V^{N+1}}{\Lambda^{3(N+1)} N!} \int_0^1 d\lambda \int dV \exp[-\beta PV] \\ \times \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] \int d\mathbf{s}_{\text{frac}} \exp[-\beta U(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.113)$$

with corresponding probability density:

$$\mathcal{N}(N, \mathbf{s}^N, s_{\text{frac}}, \lambda) \propto \frac{\exp[\beta\mu N] V^{N+1}}{\Lambda^{3(N+1)} N!} \exp[-\beta PV] \exp[-\beta U(\mathbf{s}^N)] \exp[-\beta U(\mathbf{s}_{\text{frac}}, \mathbf{s}^N, \lambda)] \quad (6.114)$$

where N is the number of whole molecules, V is the volume of the system, P is the imposed pressure, Λ is the thermal wavelength, μ is the imposed chemical potential, \mathbf{s}^N is the scaled coordinate vector of all molecules, U is the potential energy (excluding the interactions of the fractional molecules), U_{frac} is the potential energy of the fractional molecule (including interactions between fractional molecules), $\lambda \in [0, 1]$ is the fractional parameter, and \mathbf{s}_{frac} is the scaled coordinate vector of the fractional molecule.

6.9.2 Acceptance Rules

In addition to the trial moves from the conventional *NVT* and *NPT* ensemble, there is a trial move that changes the value of the fractional parameter λ . Using Eqs. (6.4) and (6.112) the acceptance rules for the CFC Grand-Canonical Ensemble and CFC Osmotic Ensemble can be derived.

Translation, Rotation and Torsion Trial Moves

Performing these trial moves only result in an energy change of the system and therefore:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U]) \quad (6.115)$$

where ΔU is a change in the intermolecular energy for the translation and rotation trial moves. For the torsion trial move, ΔU contains the change in intramolecular energy as well:

$$\Delta U = U(\mathbf{s}_n^N) - U(\mathbf{s}_o^N) + U_{\text{frac}}(\mathbf{s}_{\text{frac},n}^{N_{\text{frac}}}, \mathbf{s}_n^N, \lambda, \delta) - U_{\text{frac}}(\mathbf{s}_{\text{frac},o}^{N_{\text{frac}}}, \mathbf{s}_o^N, \lambda, \delta). \quad (6.116)$$

Volume Changes

For a volume change (only in the Osmotic Ensemble) the acceptance rule is similar to that in the CFC*NPT* ensemble:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \left(\frac{V_n}{V_o} \right)^{N+N_{\text{frac}}} \exp[-\beta P(V_n - V_o)] \exp[-\beta\Delta U] \right) \quad (6.117)$$

$$= \min \left(1, \exp[-\beta P(V_n - V_o) - \beta\Delta U + (N + N_{\text{frac}}) \ln \left(\frac{V_n}{V_o} \right)] \right) \quad (6.118)$$

where it should be noted that ΔU only depends on the change in volume because of the rescaling of the coordinates \mathbf{r} and \mathbf{r}_{frac} :

$$\Delta U = U_n(\mathbf{s}^N) - U_o(\mathbf{s}^N) + U_{\text{frac},n}(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta) - U_{\text{frac},o}(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda, \delta) \quad (6.119)$$

Lambda Trial Move

The lambda trial move in the Grand-Canonical and Osmotic Ensemble is different than the one from the conventional CFC ensembles. Here, a lambda trial move changes the value of λ ($\lambda_n = \lambda_o + \Delta\lambda$) but now λ_n is not restricted to the interval $(0, 1)$. There are three cases to consider:

I. $\lambda_n < 0$

In this case, the fractional molecule is removed from the system and a randomly selected molecule of the same type becomes the new fractional molecule. (If no molecules are present to transform into a new fractional the whole trial move is rejected immediately.) The new fractional parameter is $\lambda_n \rightarrow \lambda_n + 1$ (such that it is in the interval $(0, 1)$). This acceptance rule is as follows:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{N_o}{f\beta V} \exp[-\beta\Delta U] \right) \quad (6.120)$$

where f is the fugacity, $N_o = N_n + 1$ and, ΔU the total energy difference between the old and the new configuration.

II. $0 < \lambda_n < 1$

In this case, the lambda trial move is treated similar to the trial moves in the CFCNVT and CFCNPT ensemble. The acceptance rule is:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U]) \quad (6.121)$$

where ΔU depends only on the difference in λ because the positions of the molecules are unchanged:

$$\Delta U = U_{\text{frac}} \left(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda_n, \delta \right) - U_{\text{frac}} \left(\mathbf{s}_{\text{frac}}^{N_{\text{frac}}}, \mathbf{s}^N, \lambda_o, \delta \right). \quad (6.122)$$

III. $1 < \lambda_n$

In this case, the fractional molecule is transformed into a normal molecule and a new fractional molecule is added to the system at random position. The new fractional parameter is $\lambda_n \rightarrow \lambda_n - 1$ (such that it is in the interval $(0, 1)$). This acceptance rule is as follows:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{f\beta V}{N_o + 1} \exp[-\beta\Delta U] \right) \quad (6.123)$$

where f is the fugacity, $N_o = N_n - 1$ and, ΔU the total energy difference between the old and the new configuration.

Chapter 7

Trial Moves

In this section we briefly explain the available Monte Carlo trial moves. Acceptance rules are derived in Chapter 6.

7.1 Translation

7.1.1 Normal

The algorithm for the normal translation trial move works as follows:

1. Select a random molecule and calculate its energy E_o .
2. Give the molecule a random displacement in a random direction (for example in the x -direction):

$$x \rightarrow x + \Delta x \quad (7.1)$$

such that $\Delta x < \Delta x_{\max}$.

3. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
4. Accept the trial move according to the acceptance rule

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta \Delta U]). \quad (7.2)$$

The maximum displacement Δx_{\max} depends on the type of molecule and the box it is in. During the initializing and equilibrating phase of the simulation, the maximum displacements are modified to obtain a 50% acceptance ratio. See Figure 7.1 for a illustration of this trial move.

7.1.2 Pair

The algorithm for the pair translation trial move [43] works as follows and is only performed in the vapor phase box:

1. Select a random molecule type pair: type i and type j .

2. Select a random molecule of type i in the vapor phase.
3. Select a molecule of type j with probability distribution:

$$P_j^o = \frac{w(r_{ij})}{\sum_k w(r_{ik})}, \quad w(r_{ij}) = \exp \left[\frac{\beta}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \right]. \quad (7.3)$$

4. Calculate the energy of the two molecules E_o .
5. Displace both molecules by the same displacement in a random direction (for example in the x -direction):

$$x_{i/j} \rightarrow x_{i/j} + \Delta x \quad (7.4)$$

such that $\Delta x < \Delta x_{\max}$.

6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
7. Calculate P_j^n for the new configuration.
8. Accept the trial move according to the acceptance rule

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{P_j^n}{P_j^o} \exp[-\beta \Delta U] \right). \quad (7.5)$$

The maximum displacement Δx_{\max} is the average of the two maximum displacements for a normal translation trial move for the two molecule types.

7.1.3 Cluster

The algorithm for the cluster translation trial move works as follows and is only performed in the vapor phase box:

1. Determine all clusters in the box according to the distance between molecules. Molecules are in the same cluster if the distance between their centers of mass is less than some value R_{cluster} .
2. Select a random cluster. (If there is only 1 cluster, no trial move is performed.)
3. Calculate the energy of the molecules in the cluster E_o .
4. Displace all molecules in the cluster by the same displacement in a random direction (for example in the x -direction):

$$x_{i=1,N} \rightarrow x_{i=1,N} + \Delta x \quad (7.6)$$

such that $\Delta x < \Delta x_{\max}$.

5. Check if new clusters are formed by this translation, if so, then reject the trial move since it would violate detailed balance.

6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
7. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]). \quad (7.7)$$

The maximum displacement Δx_{\max} is the average of the maximum displacements for a normal translation trial move of the molecules in the cluster.

7.1.4 Smart

Typical translation trial moves in Monte Carlo simulations attempt to translate one molecule in a random direction by a random distance. One can optimize the efficiency of these trial moves by using the force on the particles. Instead of displacing one molecule at random, in Smart Monte Carlo simulations [26] all molecules are translated at the same time where the direction and distance are determined by the force and a random contribution [44]. This results in a faster equilibration of the system. This method is similar to the hybrid Monte Carlo method [45] that combines Monte Carlo and Molecular Dynamics. The Smart Monte Carlo translation trial move follows the following procedure [26]:

1. Random vectors \mathbf{u}_i are generated, where the vector components are drawn from a normal distribution with the mean equal to 0 and the variance equal to 1.
2. The 'effective kinetic energy' is calculated:

$$\kappa = \sum_{i=1}^N \frac{1}{2} m_i \mathbf{u}_i^2$$

where m_i is the mass of molecule i .

3. Each particle in the system is assigned a velocity

$$\mathbf{v}_{i,\text{old}} = \sqrt{\frac{3Nk_B T}{2\kappa}} \cdot \mathbf{u}_i$$

where the factor $\sqrt{\frac{3Nk_B T}{2\kappa}}$ guarantees that the kinetic energy of the system is equal to $\frac{3}{2} N k_B T$ (according to the equipartition theorem).

4. The forces \mathbf{F}_i acting on each molecule are calculated.

5. Velocities are updated:

$$\mathbf{v}_i = \mathbf{v}_{i,\text{old}} + \frac{\mathbf{F}_i}{2m_i} \Delta t$$

6. Positions are updated:

$$\mathbf{r}_{i,\text{new}} = \mathbf{r}_{i,\text{old}} + \mathbf{v}_i \Delta t$$

7. The forces \mathbf{F}_i are recalculated (based on the new positions $\mathbf{r}_{i,\text{new}}$).

8. Velocities are updated again:

$$\mathbf{v}_{i,\text{new}} = \mathbf{v}_i + \frac{\mathbf{F}_i}{2m_i} \Delta t$$

9. Repeat steps 4 to 8 N_{step} times (N_{step} is the number of times the equations of motion are integrated).
10. The trial move is accepted or rejected according to the acceptance rule:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta(\Delta U + \Delta K)])$$

where ΔU is the difference in potential energy between the new and old configuration:

$$\Delta U = U(\mathbf{r}_{\text{new}}^N) - U(\mathbf{r}_{\text{old}}^N),$$

and ΔK is the difference in kinetic energy:

$$\Delta K = \sum_{i=1}^N \frac{1}{2} m_i (\mathbf{v}_{i,\text{new}}^2 - \mathbf{v}_{i,\text{old}}^2)$$

In this routine, the velocity Verlet algorithm [46] is used to calculate the velocities and positions of the particles. This algorithm is time reversible and area-preserving (symplectic) [47]. The time step Δt and the number of time steps N_{step} can be adjusted to optimize the efficiency. Choosing a small time step results in small displacements and a high acceptance probability of this trial move. However, small displacements are not that useful for equilibration of the system. Choosing a large time step results in large displacements and small acceptance probabilities. In a similar way, choosing a small number of steps results in small displacements and high acceptance probabilities, while a large number of steps results in large displacements and small acceptance probabilities. One has to tune Δt and N_{step} for efficiency. In Fig. 7.2 the average accepted displacement and acceptance ratio as a function of the time step is shown for the Smart Monte Carlo translation trial move, with only one step ($N_{\text{step}} = 1$) in a system of Lennard-Jones particles.

7.2 Rotation

7.2.1 Normal

The algorithm for the normal rotation trial move works as follows:

1. Select a random molecule and calculate its energy E_{o} .
2. Rotate the molecule with a random angle $\Delta\theta \in (-\Delta\theta_{\text{max}}, \Delta\theta_{\text{max}})$ around the x -, y - or z -axis (randomly selected) and its center of mass.
3. Calculate the new energy E_{n} and $\Delta U = E_{\text{n}} - E_{\text{o}}$.
4. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U]). \quad (7.8)$$

The maximum rotation angle $\Delta\theta_{\text{max}}$ depends on the type of molecule and the box it is in. During the initializing and equilibrating phase of the simulation, the maximum rotation angles are modified to obtain a 50% acceptance ratio. See Figure 7.3 for a illustration of this trial move.

7.2.2 Pair

The algorithm for the pair rotation trial move [43] works as follows and is only performed in the vapor phase box:

1. Select a random molecule type pair: type i and type j .
2. Select a random molecule of type i in the vapor phase.
3. Select a molecule of type j with probability distribution:

$$P_j^o = \frac{w(r_{ij})}{\sum_k w(r_{ik})}, \quad w(r_{ij}) = \exp \left[\frac{\beta}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \right]. \quad (7.9)$$

4. Rotate the molecule pair with a random angle $\Delta\theta \in (-\Delta\theta_{\max}, \Delta\theta_{\max})$ around the x -, y - or z -axis and its common center of mass.
5. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
6. Calculate P_j^n for the new configuration.
7. Accept the trial move according to the acceptance rule

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{P_j^n}{P_j^o} \exp[-\beta\Delta U] \right). \quad (7.10)$$

The maximum rotation angle $\Delta\theta_{\max}$ is the average of the two maximum rotation angles for a normal rotation move for the two molecule types.

7.2.3 Cluster

The algorithm for the cluster rotation trial move works as follows and is only performed in the vapor phase box:

1. Determine all clusters in the box according to the distance between molecules. Molecules are in the same cluster if the distance between their centers of mass is less than some value R_{cluster} .
2. Select a random cluster. (If there is only 1 cluster, no trial move is performed.)
3. Calculate the energy of the molecules in the cluster E_o .
4. Rotate all molecules in the cluster with a random angle $\Delta\theta \in (-\Delta\theta_{\max}, \Delta\theta_{\max})$ around the x -, y - or z -axis and its common center of mass.
5. Check if new clusters are formed by this rotation, if so, than reject the trial move since it would violate detailed balance.
6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.

- Accept the trial move with the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]). \quad (7.11)$$

The maximum rotation angle $\Delta\theta_{\max}$ is the average of the maximum rotation angles for a normal rotation trial move of the molecules in the cluster.

7.2.4 Smart

Similar to the smart translation trial moves in Sec. 7.1.4, one can define smart rotation trial moves. In these trial moves, collective rotation of molecules as rigid bodies is performed using a short MD simulation in the *NVE* ensemble. The time step size (Δt) and trajectory length (N_{step}) of this MD run are chosen to maximize the efficiency of the sampling. For the rigid body rotation of molecules, the velocity Verlet-based algorithm of Miller and coworkers (NOSQUISH) is used [48], which is symplectic and time reversible. At every time step, all molecules are rotated according to the total torque acting on the molecules, and only intermolecular interactions are taken into account to compute the forces and torques. For more details about the implementation of the algorithm, we refer the reader to Ref. [7]. For a discussion on rigid-body integrators we would like to refer the reader to the user manual of DL_Poly [49] as well as Ref. [50].

7.3 Volume Change

7.3.1 Normal

There are three variants for the normal volume change trial move: in the *NPT* ensemble, the Gibbs Ensemble with constant total volume and the Gibbs Ensemble with constant pressure.

NPT

The algorithm for the normal volume change trial move in the *NPT* ensemble is the same in the conventional ensemble as in the expanded CFC ensemble (N is the total number of molecules including the fractional molecules). The algorithm works as follows:

- Calculate the energy of all molecules in the box E_o .
- Change the volume of the box $V_o \rightarrow V_o + \Delta V = V_n$, where $\Delta V \in (-\Delta V_{\max}, \Delta V_{\max})$.
- Scale the centers of mass of all molecules in the system according to the new volume:

$$\begin{aligned} x_{\text{CM}} &\rightarrow \alpha x_{\text{CM}} \\ y_{\text{CM}} &\rightarrow \alpha y_{\text{CM}} \\ z_{\text{CM}} &\rightarrow \alpha z_{\text{CM}} \end{aligned} \quad \alpha = \left(\frac{V_n}{V_o} \right)^{\frac{1}{3}}$$

Interatomic distances in molecules are unchanged this way.

- Calculate the new energy E_n and $\Delta U = E_n - E_o$.

- Accept the trial move with the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U - \beta P(V_{\text{n}} - V_{\text{o}}) + N \ln(V_{\text{n}}/V_{\text{o}})]). \quad (7.12)$$

The maximum change in volume ΔV_{\max} is modified during the initializing phase to obtain a 50% acceptance ratio. See Figure 7.4 for an illustration of this trial move.

Gibbs Ensemble at Constant Volume

The algorithm for the normal volume change trial move in the Gibbs Ensemble at constant total volume (N_i is the number of molecules in box i , including the fractional molecules) works as follows:

- Calculate the total energy of all molecules in both boxes E_{o} .
- Change the volume of box 1 and box 2

$$\begin{aligned} V_{1,\text{o}} &\rightarrow V_{1,\text{o}} + \Delta V = V_{1,\text{n}} \\ V_{2,\text{o}} &\rightarrow V_{2,\text{o}} + \Delta V = V_{2,\text{n}} \end{aligned} \quad (7.13)$$

where $\Delta V \in (-\Delta V_{\max}, \Delta V_{\max})$.

- Scale the centers of mass of all molecules in the system according to the new volumes (of the boxes they are in):

$$\begin{aligned} x_{\text{CM}} &\rightarrow \alpha x_{\text{CM}} \\ y_{\text{CM}} &\rightarrow \alpha y_{\text{CM}} \quad \alpha = \left(\frac{V_{\text{n}}}{V_{\text{o}}}\right)^{\frac{1}{3}} \\ z_{\text{CM}} &\rightarrow \alpha z_{\text{CM}} \end{aligned} \quad (7.14)$$

Interatomic distances in molecules are unchanged this way.

- Calculate the new total energy E_{n} and $\Delta U = E_{\text{n}} - E_{\text{o}}$.
- Accept the trial move with the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min\left(1, \frac{V_{1,\text{n}}^{N_1} V_{2,\text{n}}^{N_2}}{V_{1,\text{o}}^{N_1} V_{2,\text{o}}^{N_2}} \exp[-\beta\Delta U]\right). \quad (7.15)$$

The maximum change in volume ΔV_{\max} is modified during the initializing phase to obtain a 50% acceptance ratio. See Figure 7.5 for an illustration of this trial move.

Gibbs Ensemble at Constant Pressure

The algorithm for the normal volume change trial move in the Gibbs Ensemble with constant pressure (N_i is the number of molecules in box i including the fractional molecules) is as follows:

- Select one box at random (box i).
- Calculate the total energy of all molecules in this box E_{o} .

3. Change the volume of the box

$$V_{i,o} \rightarrow V_{i,o} + \Delta V = V_{i,n} \quad (7.16)$$

where $\Delta V \in (-\Delta V_{\max}, \Delta V_{\max})$.

4. Scale the center of masses of all molecules in the selected box according to the new volume:

$$\begin{aligned} x_{CM} &\rightarrow \alpha x_{CM} \\ y_{CM} &\rightarrow \alpha y_{CM} \quad \alpha = \left(\frac{V_n}{V_o} \right)^{\frac{1}{3}} \\ z_{CM} &\rightarrow \alpha z_{CM} \end{aligned} \quad (7.17)$$

Interatomic distances in molecules are unchanged this way.

5. Calculate the new energy E_n in the box and $\Delta U = E_n - E_o$.

6. Accept the trial move with the acceptance rule

$$\text{acc}(o \rightarrow n) = \min (1, \exp[-\beta \Delta U - \beta P(V_{i,n} - V_{i,o}) + N_i \ln(V_{i,n}/V_{i,o})]). \quad (7.18)$$

The maximum change in volume ΔV_{\max} is modified during the initializing phase to obtain a 50% acceptance ratio. See Figure 7.4 for an illustration of this trial move (for one simulation box).

7.3.2 Cluster

The volume change trial moves for clusters are similar to the conventional ones.

NPT

In the *NPT* ensemble a cluster volume change works the same as the normal cluster volume change but with $N = N_{\text{cluster}}$.

1. Determine all clusters in the vapor box according to the distance between molecules. Molecules are in the same cluster if the distance between their center of masses is less than some value R_{cluster} . The number of clusters is N_{cluster} .
2. Calculate the energy of all molecules in the box E_o .
3. Change the volume of the box $V_o \rightarrow V_o + \Delta V = V_n$, where $\Delta V \in (-\Delta V_{\max}, \Delta V_{\max})$.
4. Scale the centers of mass of all clusters in the system according to the new volume:

$$\begin{aligned} x_{CM,\text{cluster}} &\rightarrow \alpha x_{CM,\text{cluster}} \\ y_{CM,\text{cluster}} &\rightarrow \alpha y_{CM,\text{cluster}} \quad \alpha = \left(\frac{V_n}{V_o} \right)^{\frac{1}{3}} \\ z_{CM,\text{cluster}} &\rightarrow \alpha z_{CM,\text{cluster}} \end{aligned} \quad (7.19)$$

Interatomic distances in molecules and intermolecular distances in a cluster are unchanged this way.

5. Check if new clusters are formed by the volume change, if so, then reject the trial move since it would violate detailed balance.
6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
7. Accept the trial move with the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U - \beta P(V_n - V_o) + N_{\text{cluster}} \ln(V_n/V_o)]). \quad (7.20)$$

The maximum change in volume ΔV_{\max} is modified during the initializing phase to obtain a 50% acceptance ratio.

Gibbs Ensemble at Constant Volume

The algorithm for the cluster volume change trial move in the Gibbs Ensemble at constant total volume (N_i is the number of molecules in box i , fractionals included) is as follows:

1. Determine all clusters in the vapor box (for example box 2) according to the distance between molecules. Molecules are in the same cluster if the distance between their centers of mass is less than some value R_{cluster} . The number of clusters is N_{cluster} .
2. Calculate the total energy of all molecules in both boxes E_o .
3. Change the volume of box 1 and box 2

$$\begin{aligned} V_{1,o} &\rightarrow V_{1,o} + \Delta V = V_{1,n} \\ V_{2,o} &\rightarrow V_{2,o} + \Delta V = V_{2,n} \end{aligned} \quad (7.21)$$

where $\Delta V \in (-\Delta V_{\max}, \Delta V_{\max})$.

4. For the liquid phase (box 1), the coordinates are scaled in the conventional way:

$$\begin{aligned} x_{\text{CM}} &\rightarrow \alpha x_{\text{CM}} \\ y_{\text{CM}} &\rightarrow \alpha y_{\text{CM}} \quad \alpha = \left(\frac{V_{1,n}}{V_{1,o}}\right)^{\frac{1}{3}} \\ z_{\text{CM}} &\rightarrow \alpha z_{\text{CM}} \end{aligned} \quad (7.22)$$

Interatomic distances in molecules are unchanged this way. For the vapor phase (box 2), scale the centers of mass of all clusters in the system according to the new volume:

$$\begin{aligned} x_{\text{CM,cluster}} &\rightarrow \alpha x_{\text{CM,cluster}} \\ y_{\text{CM,cluster}} &\rightarrow \alpha y_{\text{CM,cluster}} \quad \alpha = \left(\frac{V_{2,n}}{V_{2,o}}\right)^{\frac{1}{3}} \\ z_{\text{CM,cluster}} &\rightarrow \alpha z_{\text{CM,cluster}} \end{aligned} \quad (7.23)$$

Interatomic distances in molecules and intermolecular distances in a cluster are unchanged this way.

5. Check if new clusters are formed by the volume change in the vapor phase, if so, then reject the trial move since it would violate detailed balance.
6. Calculate the new total energy E_n and $\Delta U = E_n - E_o$.

7. Accept the trial move with the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{V_{1,\text{n}}^{N_1} V_{2,\text{n}}^{N_{\text{cluster}}}}{V_{1,\text{o}}^{N_1} V_{2,\text{o}}^{N_{\text{cluster}}}} \exp[-\beta \Delta U] \right). \quad (7.24)$$

The maximum change in volume ΔV_{\max} is modified during the initializing phase to obtain a 50% acceptance ratio.

Gibbs Ensemble at Constant Pressure

The algorithm for the cluster volume change trial move in the Gibbs Ensemble at constant pressure (N_i is the number of molecules in box i , fractionals included) is as follows:

1. Select one box at random (box i).
2. If this box is the liquid phase than perform a normal NPT volume change trial move.
3. Otherwise, determine all clusters in the vapor box according to the distance between molecules. Molecules are in the same cluster if the distance between their centers of mass is less than some value R_{cluster} . The number of clusters is N_{cluster} .
4. Calculate the total energy of all molecules in this box E_{o} .
5. Change the volume of the box

$$V_{i,\text{o}} \rightarrow V_{i,\text{o}} + \Delta V = V_{i,\text{n}} \quad (7.25)$$

where $\Delta V \in (-\Delta V_{\max}, \Delta V_{\max})$

6. Scale the centers of mass of all clusters in the system according to the new volume:

$$\begin{aligned} x_{\text{CM,cluster}} &\rightarrow \alpha x_{\text{CM,cluster}} \\ y_{\text{CM,cluster}} &\rightarrow \alpha y_{\text{CM,cluster}} \quad \alpha = \left(\frac{V_{\text{n}}}{V_{\text{o}}} \right)^{\frac{1}{3}} \\ z_{\text{CM,cluster}} &\rightarrow \alpha z_{\text{CM,cluster}} \end{aligned} \quad (7.26)$$

Interatomic distances in molecules and intermolecular distances in a cluster are unchanged this way.

7. Check if new clusters are formed by the volume change, if so, than reject the move since it would violate detailed balance.
8. Calculate the new energy E_{n} and $\Delta U = E_{\text{n}} - E_{\text{o}}$.
9. Accept the trial move with the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min (1, \exp[-\beta \Delta U - \beta P (V_{i,\text{n}} - V_{i,\text{o}}) + N_{\text{cluster}} \ln (V_{i,\text{n}}/V_{i,\text{o}})]). \quad (7.27)$$

The maximum change in volume ΔV_{\max} is modified during the initializing phase to obtain a 50% acceptance ratio.

7.4 Intramolecular

7.4.1 Bending

A bond angle in a molecule is changed in this trial move. The algorithm works as follows.

1. Select a random molecule and calculate E_o .
2. Randomly select a bond angle (θ_o) in this molecule.
3. Bend a part of the molecule along a bond that defines the bond bending with a random angle $\Delta\theta \in (-\Delta\theta_{\max}, \Delta\theta_{\max})$:

$$\theta_n = \theta_o + \Delta\theta \quad (7.28)$$

4. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
5. Accept the trial move according to the acceptance rule

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{\sin(\theta_n)}{\sin(\theta_o)} \exp[-\beta\Delta U] \right). \quad (7.29)$$

The maximum rotation angle $\Delta\theta_{\max}$ depends on the type of molecule. It is **not** optimized to achieve a 50% acceptance ratio since bond bendings of the same type can behave very different depending on their position in the molecule. See Figure 7.6 for an illustration of this trial move.

7.4.2 Torsion Move

A torsion angle in a molecule is changed in this trial move. The algorithm works as follows.

1. Select a random molecule and calculate E_o .
2. Randomly select a torsion angle in this molecule.
3. Rotate part of the molecule along a bond that defines the torsion with a random angle $\Delta\phi \in (-\Delta\phi_{\max}, \Delta\phi_{\max})$.
4. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
5. Accept the trial move according to the acceptance rule

$$\text{acc}(o \rightarrow n) = \min (1, \exp[-\beta\Delta U]). \quad (7.30)$$

The maximum rotation angle $\Delta\phi_{\max}$ depends on the type of molecule. It is **not** optimized to achieve a 50% acceptance ratio since torsions of the same type can behave very different depending on their position in the molecule. See Figure 7.7 for an illustration of this trial move.

7.5 Lambda Moves

7.5.1 Normal

The value of the fractional parameter λ is changed in this trial move. The algorithm works as follows.

1. Select a random fractional group and calculate the energy E_o .
2. Change the value of λ :

$$\lambda \rightarrow \lambda + \Delta\lambda \quad (7.31)$$

where $\Delta\lambda \in (-\Delta\lambda_{\max}, \Delta\lambda_{\max})$.

3. If $\lambda < 0$ or $\lambda > 1$ reject the trial move. Otherwise, calculate the new energy E_n and $\Delta U = E_n - E_o$.
4. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta\Delta U]). \quad (7.32)$$

The maximum change $\Delta\lambda_{\max}$ is **not** optimized to achieve a 50% acceptance ratio, instead a weight function can be used to optimize the efficiency of the simulation. See Figures 7.8, 7.9 and 7.10 for illustrations of this trial move in different ensembles.

7.5.2 Grand-Canonical Ensemble

The value of the fractional parameter λ is changed in this trial move. The algorithm works as follows [42].

1. Select a random fractional and calculate the energy E_o .
2. Change the value of λ :

$$\lambda \rightarrow \lambda + \Delta\lambda \quad (7.33)$$

where $\Delta\lambda \in (-\Delta\lambda_{\max}, \Delta\lambda_{\max})$.

There are now three options:

- I. $\lambda < 0$
- II. $0 < \lambda < 1$
- III. $1 < \lambda$

and a different procedure for each option is followed.

- I. $\lambda < 0$
- 3. Delete the selected fractional molecule.

4. Select a random molecule of the same type as the deleted fractional.
5. Change the selected molecule into a fractional with the remainder $\lambda \rightarrow \lambda + 1$.
6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
7. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{f\beta V}{N+1} \exp[-\beta\Delta U] \right). \quad (7.34)$$

II. $0 < \lambda < 1$

3. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
4. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min (1, \exp[-\beta\Delta U]). \quad (7.35)$$

III. $1 < \lambda$

3. Change the selected fractional into a whole molecule.
4. Insert a fractional molecule of the same type as the old fractional at random in the system with the remainder $\lambda \rightarrow \lambda - 1$.
5. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
6. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \frac{N-1}{f\beta V} \exp[-\beta\Delta U] \right). \quad (7.36)$$

The maximum change $\Delta\lambda_{\max}$ is **not** optimized to achieve a 50% acceptance ratio, instead a weight function can be used to optimize the efficiency of the simulation. See Figures 7.11, 7.12 and 7.13 for illustrations of the three different cases in this trial move.

7.6 CFC Swap (Reinsertion)

7.6.1 NVT/NPT

When the *NVT/NPT* swap trial move is chosen, it corresponds to a random reinsertion of a fractional molecule [2]:

1. Select a random fractional group in the system.
2. From this fractional group, select a random molecule.
3. Calculate E_o and delete the molecule from the system.
4. Reinsert the fractional molecule in the same box at a random position with a random orientation.
5. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
6. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min (1, \exp[-\beta\Delta U]). \quad (7.37)$$

See Figure 7.14 for an illustration of this trial move.

7.6.2 Gibbs Ensemble

When the Gibbs Ensemble swap trial move is chosen, a fractional group is transferred from one box to the other [38]:

1. Select a random fractional group in the system.
2. Calculate the energy of the fractional molecules E_o and delete the fractional molecules from the system (in box i).
3. Reinsert the fractional molecules randomly (position and orientation) in the other box (j).
4. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
5. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \left(\frac{V_j}{V_i} \right)^M \exp[-\beta \Delta U] \right). \quad (7.38)$$

where M is the number of molecules in the fractional group.

See Figure 7.15 for an illustration of this trial move.

7.6.3 Reaction Ensemble

When the Reaction Ensemble swap trial move is chosen, fractional molecules are changed to different molecule types: a fractional reaction [1].

1. Select a random reaction from the possible reactions.
2. Select the fractional corresponding to this reaction, this will be the reactant.
3. Select a random possible product for this reaction.
4. Calculate the energy of the fractional reactants E_o and delete them from the system.
5. Reinsert fractional product molecules randomly (position and orientation).
6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
7. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \prod_{\text{Reactants}} \left(\frac{V q_i}{\Lambda_i^3} \right)^{-\nu_i} \prod_{\text{Products}} \left(\frac{V q_i}{\Lambda_i^3} \right)^{\nu_i} \exp[-\beta \Delta U] \right). \quad (7.39)$$

See Figure 7.16 for an illustration of this trial move.

7.7 CFC Identity Change

7.7.1 NVT/NPT

When the *NVT/NPT* identity change trial move is chosen, this corresponds to a identity change of a fractional with a whole molecule of the same type [2]:

1. Select a random fractional type in the system.
2. From this fractional, select a random molecule.
3. Select a random whole molecule of the same type.
4. Calculate the energy of the two selected molecules E_o .
5. Change the whole molecule into a fractional with the same λ and change the fractional into a whole.
6. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
7. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta \Delta U]). \quad (7.40)$$

See Figure 7.17 for an illustration of this trial move.

7.7.2 Gibbs Ensemble

When the Gibbs Ensemble identity change trial move is chosen, fractional molecules in one box are changed into whole molecules and whole molecules in the other box becomes the new fractional ones [38]:

1. Select a random fractional type in the system (in box i).
2. Select whole molecule(s) in the other box (j) of the same molecule type.
3. Calculate the energy of the selected molecules E_o .
4. Change the whole molecules into fractionals with the same λ and change the fractionals into a whole molecules.
5. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
6. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min\left(1, \frac{N_j}{N_i + 1} \exp[-\beta \Delta U]\right). \quad (7.41)$$

See Figure 7.18 for an illustration of this trial move.

7.7.3 Reaction Ensemble

When the Reaction Ensemble identity change trial move is chosen, fractional molecules are changed into whole molecules and whole molecules are changed into fractionals [1]:

1. Select a random reaction from the possible reactions.
2. Select the fractional corresponding to this reaction, this will be the reactant.
3. Select a random possible product for this reaction.
4. Select random whole molecules that form the product.
5. Calculate the energy of the fractional reactants and the whole products E_o .
6. Change the fractional reactants into whole molecules and the whole products into fractional molecules with the same λ .
7. Calculate the new energy E_n and $\Delta U = E_n - E_o$.
8. Accept the trial move according to the acceptance rule

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min \left(1, \prod_{\text{Reactants}} \frac{N_i!}{(N_i - \nu_i)!} \prod_{\text{Products}} \frac{N_i!}{(N_i + \nu_i)!} \exp[-\beta \Delta U] \right). \quad (7.42)$$

See Figure 7.19 for an illustration of this trial move.

7.8 CFC Hybrid Trial Moves

The CFC hybrid trial moves combine the CFC Swap and Identity Change trial moves. The trial move is chosen based on the value of λ .

1. Select a random fractional type in the system.
2. For this fractional, use its fractional parameter λ to select a trial move in the correct ensemble:
 - If $\lambda < \lambda_{\text{swap}}$ select the CFC Swap Trial Move.
 - If $\lambda > \lambda_{\text{change}}$ select the CFC Identity Change Trial Move.
 - Otherwise no trial move is selected (continue with the next MC step).

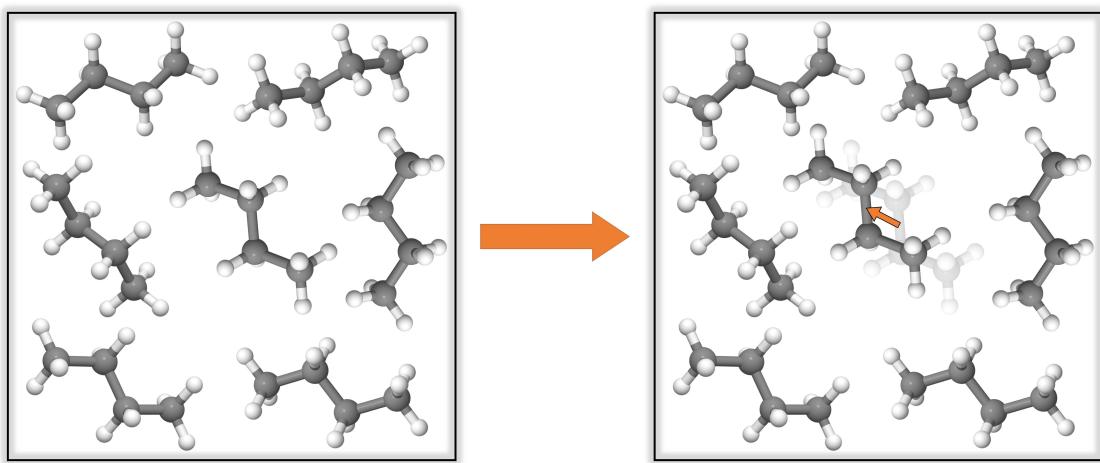


Figure 7.1: Translation Trial Move: a molecule in the system is selected at random and given a random displacement in a random direction.

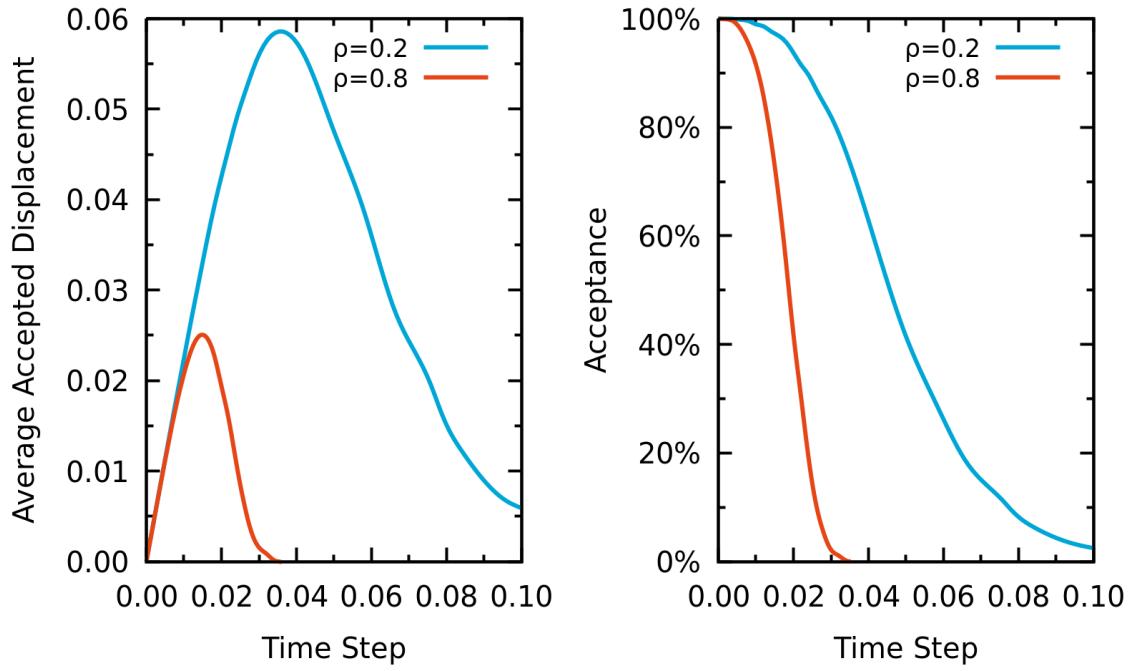


Figure 7.2: Acceptance probability and average accepted displacement of the Smart Monte Carlo translation trial move as a function of the time step in a system of Lennard-Jones particles with number densities: $\rho = 0.2$ (blue line) and $\rho = 0.8$ (red line). The temperature is set to 2.0 (reduced units). The optimal choices for Δt are around 0.04 and 0.015 (reduced units), for $\rho = 0.2$ and $\rho = 0.8$, respectively.

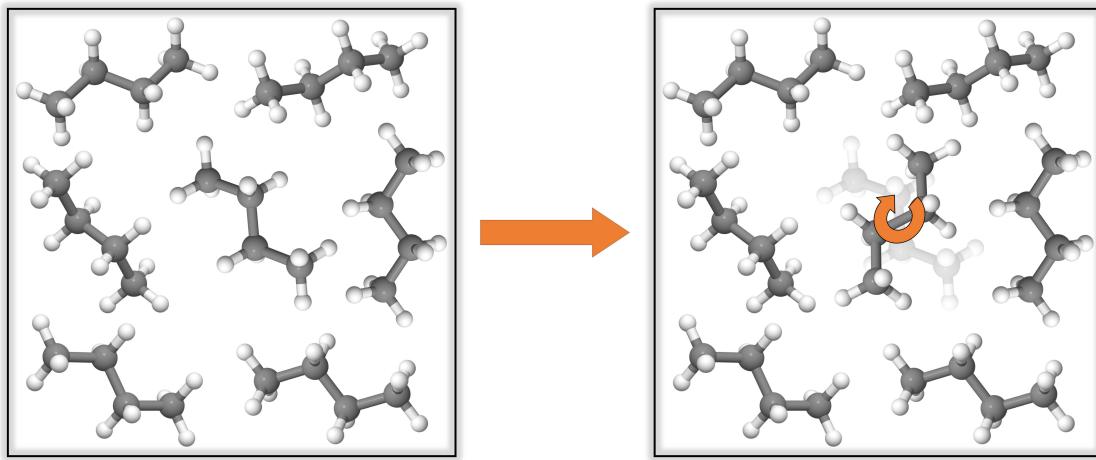


Figure 7.3: Rotation Trial Move: a molecule in the system is selected at random and given a random rotation around a random axis.

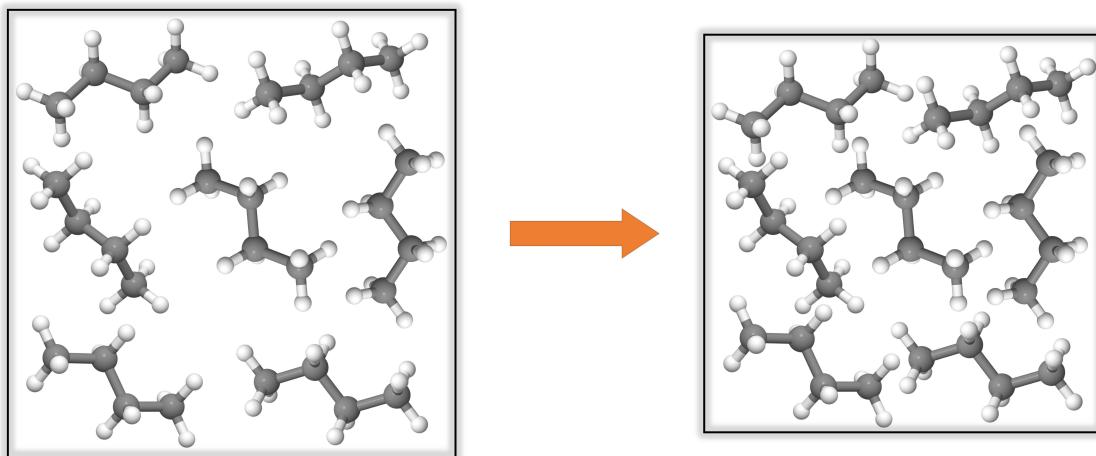


Figure 7.4: Volume Change Trial Move: the volume of the simulation box is changed by a random volume and the centers of mass of the molecules are scaled accordingly.

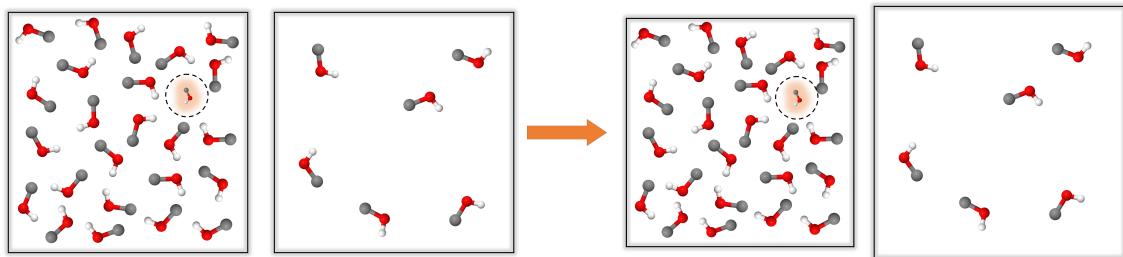


Figure 7.5: Volume Change Trial Move in the GE-*NVT* Ensemble: the volume of the simulation boxes is changed by a random volume while the total volume is conserved.

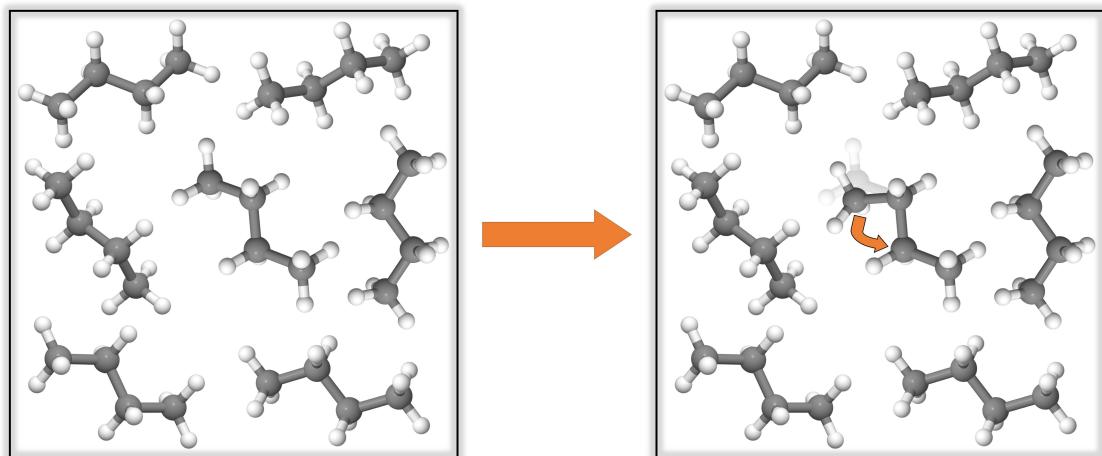


Figure 7.6: Bending Trial Move: a molecule in the system is randomly selected and a randomly selected bond angle in this molecule is changed by a random angle.

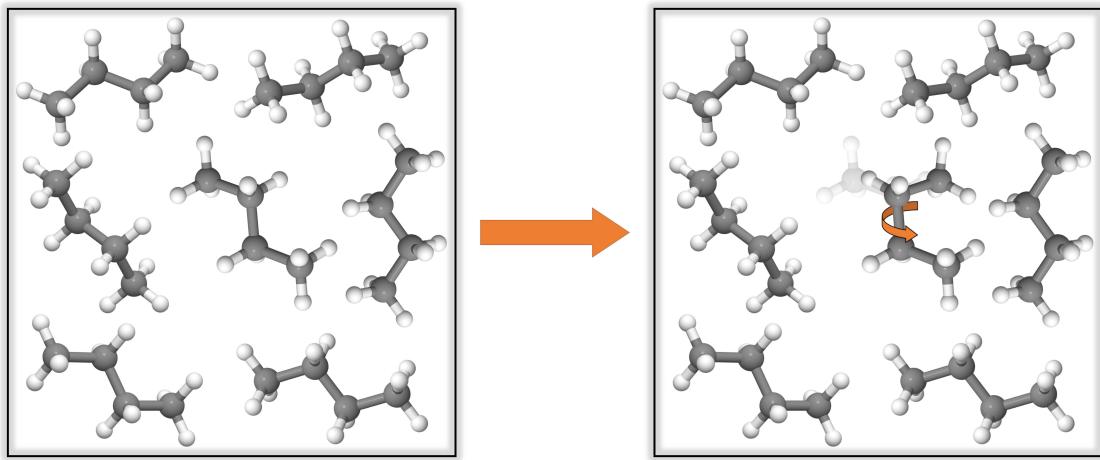


Figure 7.7: Torsion Trial Move: a molecule in the system is randomly selected and at a random position in this molecule the torsion is randomly changed.

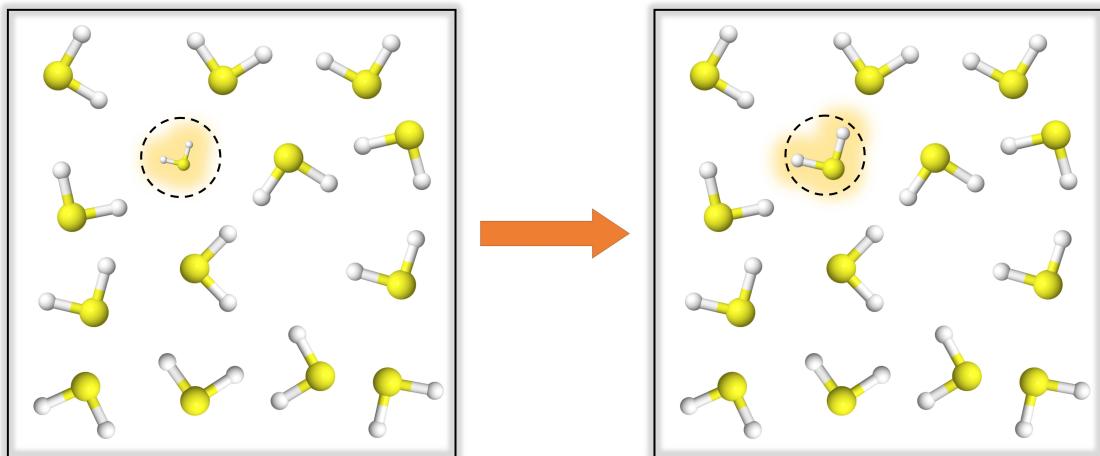


Figure 7.8: Lambda Trial Move in the NVT/NPT ensemble: a fractional group in the system is randomly selected and the value of the fractional parameter is changed by a random value.

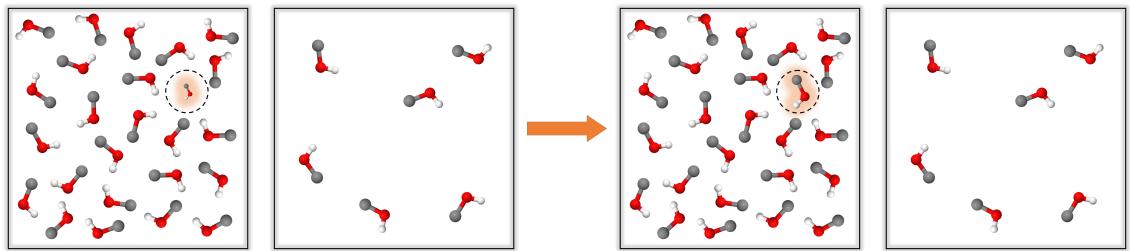


Figure 7.9: Lambda Trial Move in the Gibbs Ensemble: a fractional group in the system is randomly selected and the value of the fractional parameter is changed by a random value.

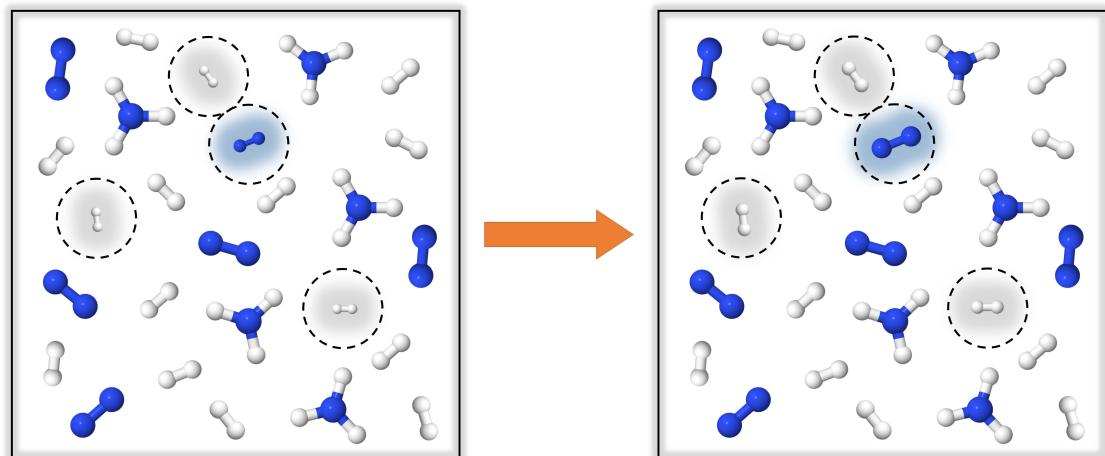


Figure 7.10: Lambda Trial Move in the Reaction Ensemble: a fractional group in the system is randomly selected and the value of the fractional parameter is changed by a random value.

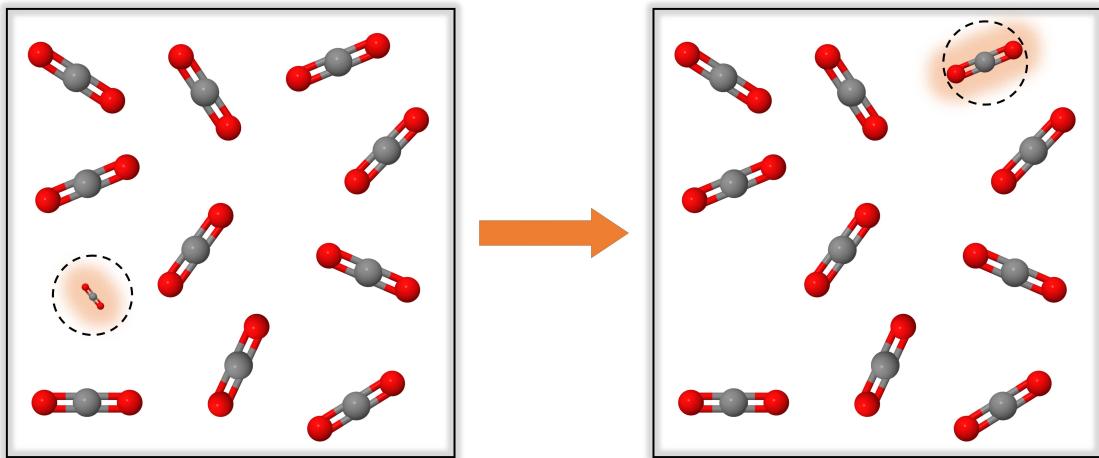


Figure 7.11: Lambda Trial Move in the Grand-Canonical Ensemble (case I): a fractional molecule in the system is randomly selected and the value of the fractional parameter is changed by a random value. The new fractional parameter is smaller than 0: the fractional molecule is removed from the system and a whole molecule is transformed into a fractional molecule.

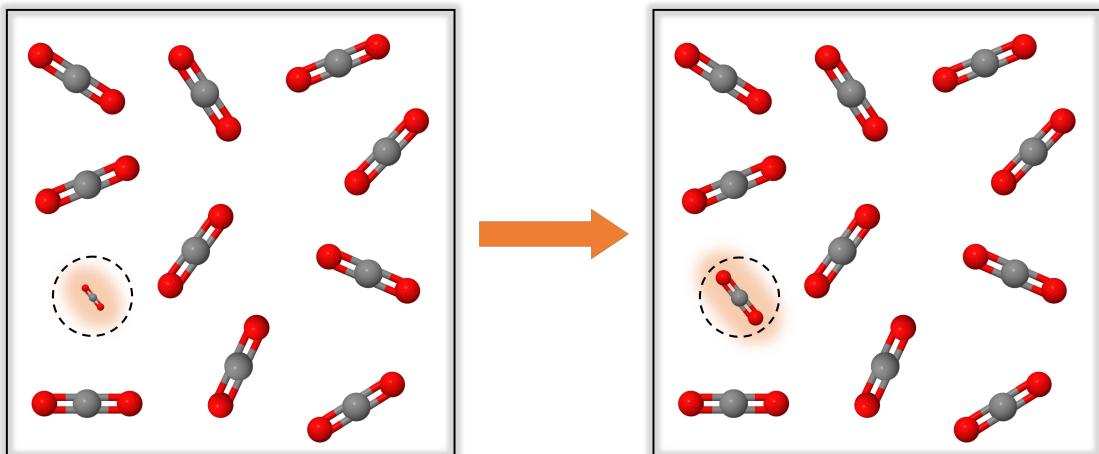


Figure 7.12: Lambda Trial Move in the Grand-Canonical Ensemble (case II): a fractional molecule in the system is randomly selected and the value of the fractional parameter is changed by a random value. The new fractional parameter is in the interval $[0, 1]$

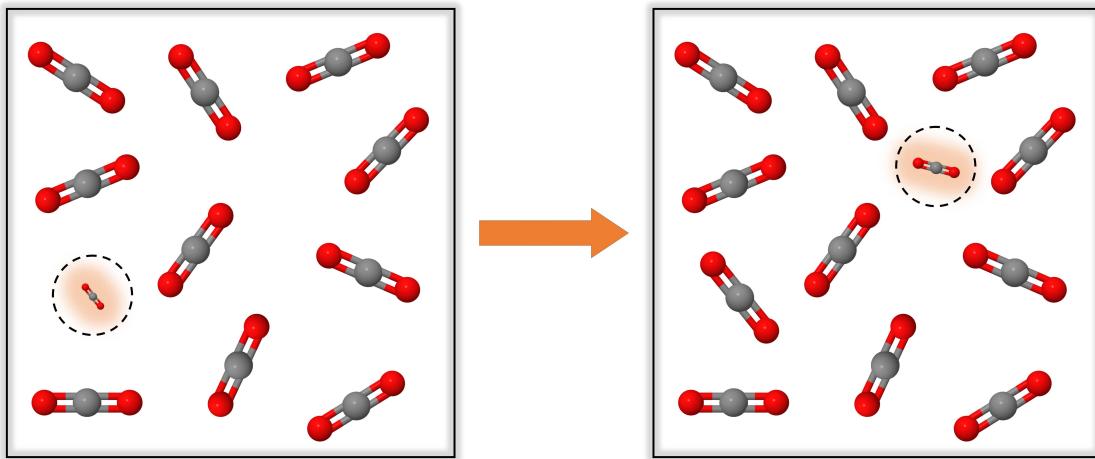


Figure 7.13: Lambda Trial Move in the Grand-Canonical Ensemble (case III): a fractional molecule in the system is randomly selected and the value of the fractional parameter is changed by a random value. The new fractional parameter is larger than 1: the fractional molecule is converted into a whole molecule and a new fractional molecule is inserted in the system.

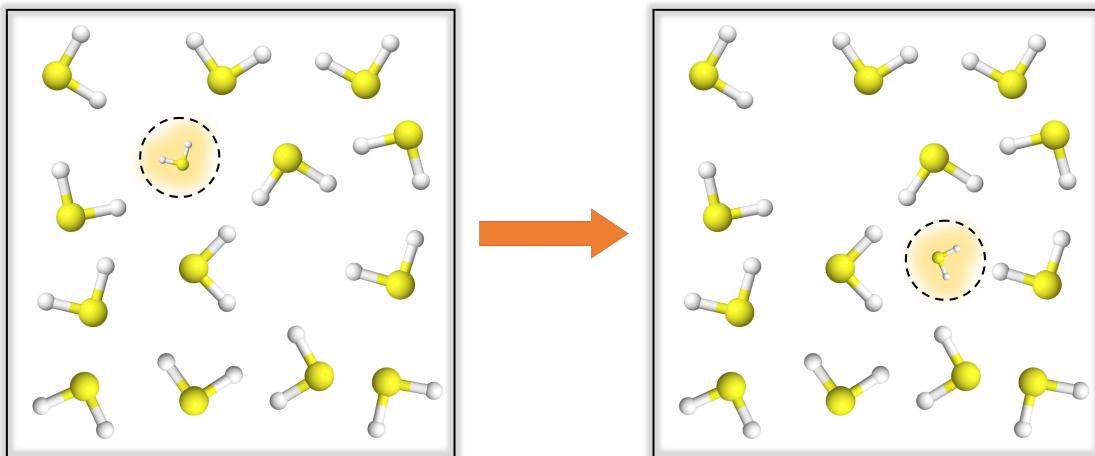


Figure 7.14: Swap Trial Move in the NVT/NPT ensemble: a fractional molecule in the system is randomly selected, removed and reinserted at a random position with a random orientation in the system.

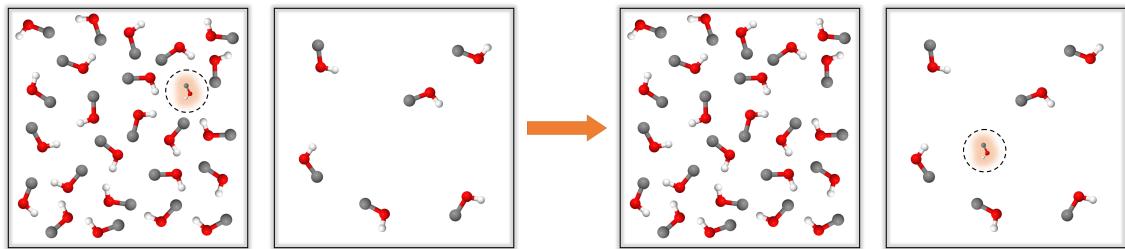


Figure 7.15: Swap Trial Move in the Gibbs Ensemble: a fractional group in the system is randomly selected, removed from the simulation box and reinserted in the other simulation box at random positions with random orientations.

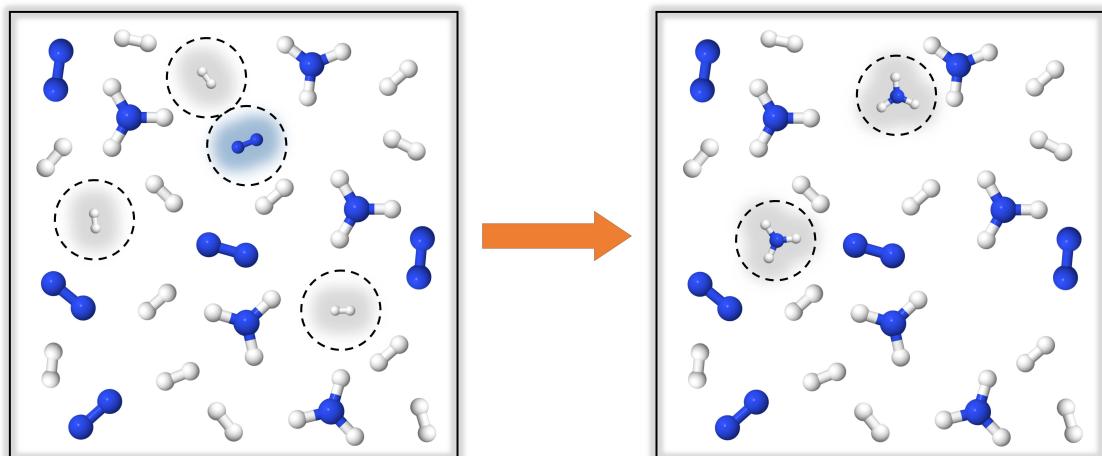


Figure 7.16: Swap Trial Move in the Reaction Ensemble: a fractional group of reactants (or products) in the system is randomly selected, removed from the simulation box and fractional molecules of products (or reactants) are inserted at random positions with random orientations.

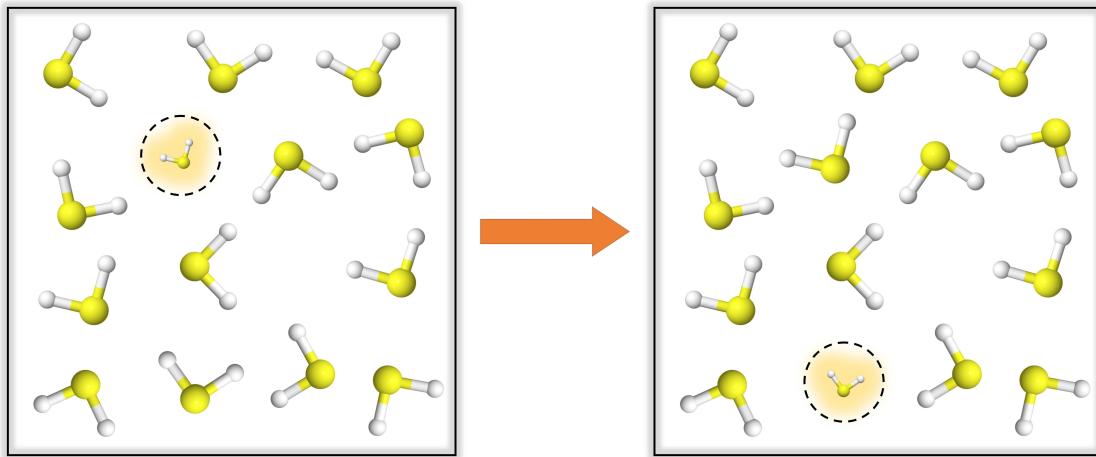


Figure 7.17: Identity Change Trial Move in the NVT/NPT ensemble: a fractional group in the system is selected at random and its molecules are converted to a whole molecules. Randomly selected whole molecules of the same molecule types are then transformed into fractional molecules.

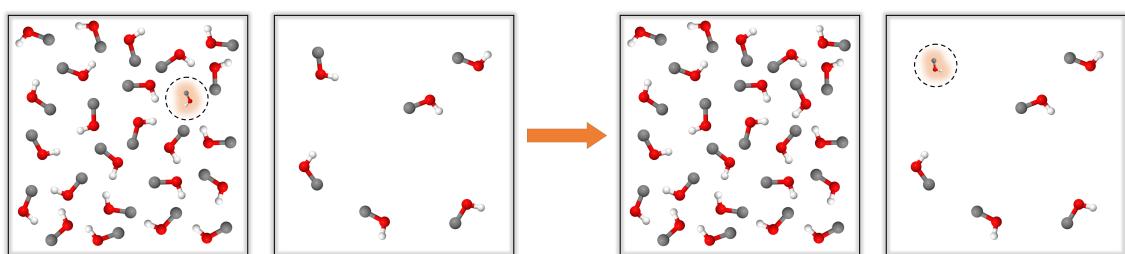


Figure 7.18: Identity Change Trial Move in the Gibbs Ensemble: a fractional group in the system is selected at random and its molecules are converted to whole molecules. Randomly selected whole molecules of the same molecule types in the other simulation box are transformed into fractional molecules.

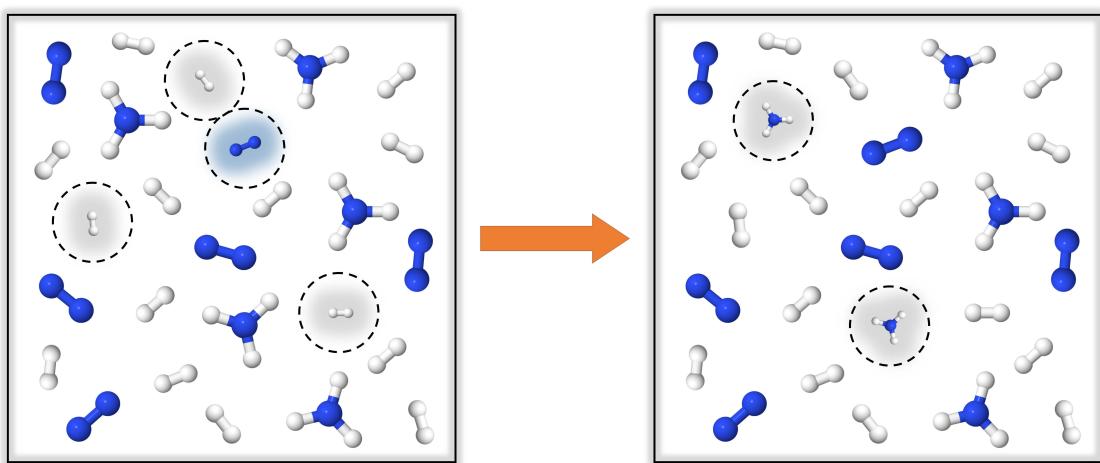


Figure 7.19: Identity Change Trial Move in the Reaction Ensemble: a fractional group of reactants (or products) in the system is randomly selected and its fractional molecules are transformed into whole molecules. Randomly selected whole molecules of products (or reactants) are transformed into fractional molecules.

Chapter 8

Examples

In the home directory of Brick-CFCMC, the folder `EXAMPLES` contains examples of simulation input files can be found. Some of them can be run relatively quick (a few minutes) and can give a brief demonstration of the different aspects of Brick-CFCMC. Other examples take much longer to run and can be used as an example of input files for more complex simulations.

8.1 *NPT* Ensemble

Example 1: Methane

In this example, only translation and volume change trial moves are used. The input files are provided for methane at 175 K and 2793 kPa. The density from original simulations with the TraPPE force field is 297 kg/m³ [18].

Example 2: Ethane (rigid)

In this example, in addition to example 1, rotation trial moves are used. The input files are provided for ethane at 256 K and 1740 kPa. The density from original simulations with the TraPPE force field is 432 kg/m³ [18].

Example 3: Propane (flexible)

In this example, in addition to example 2, bond bending trial moves are used. The input files are provided for propane at 312 K and 1519 kPa. The density from original simulations with the TraPPE force field is 467 kg/m³ [18].

Example 4: Butane (flexible)

In this example, in addition to example 3, torsion trial moves are used. The input files are provided for butane at 327 K and 754 kPa. The density from original simulations with the TraPPE force field is 536 kg/m³ [18].

Example 5: Ethanol (flexible)

In this example, smart translation trial moves along with the traditional translation and rotation trial moves are used. The input files are provided for ethanol at 425 K and 1110 kPa. The density from original simulations with the TraPPE force field is 634 kg/m³ [18].

Example 6: Water (rigid)

In this example, smart translation trial moves along with the traditional translation and rotation trial moves are used. The input files are provided for water at 298.15 K and 101.325 kPa. The density from original simulations with the TIP3P force field is 982 kg/m³ [51].

Example 7: Ideal Gas

In this example, an ideal gas is simulated. The ideal gas behavior is achieved by using the appropriate option in the `forcefield.in` file. The input files outputs are provided for an ideal gas at 200 K and 1600 kPa. The density obtained from the ideal gas equation is 962 kg/m³.

Example 8: Partial molar properties of a LJ fluid

In this example, the partial molar enthalpy and partial molar volume (derivatives of the chemical potential) for a LJ mixture are computed in the CFC *NPT* ensemble at reduced temperature of 2 and reduced pressure 6.5. The method is explained in [2]. NOTE: longer simulations are required for accurate results.

Example 9: Partial molar properties of a H₂O-CH₃OH mixture

In this example, the equimolar mixture of water and methanol in the liquid phase at 298.15 K and 1 bar is studied. By simulating pure water and pure methanol in the liquid phase (using the same parameters for electrostatics), the excess enthalpy of mixing can be calculated. For comparison, chemical potentials and activity coefficients of water and methanol are reported in [2, 33]. NOTE: in this example the cutoff is different from the one used in the paper which slightly alters the results, and also molecules are not rigid.

Example 10: Derivative of the fractional particle's energy as a function of λ .

In this example, the derivative of the energy with respect to λ is calculated for a fractional LJ particle in a box with LJ particles at a reduced temperature of 2 and a reduced pressure of 1. Integrating this partial derivative results in the excess chemical potential.

8.2 Gibbs Ensemble

Example 1: Vapor-Liquid Equilibrium of Lennard-Jones particles

In this example, the coexistence densities and chemical potentials at VLE are calculated for LJ particles in the Gibbs ensemble (constant volume). The coexistence densities and chemical potentials

are reported in Ref. [38].

Example 2: Vapor-Liquid Equilibrium of Methanol

In this example, the coexistence densities of methanol at 300 K are calculated in the Gibbs Ensemble with constant total volume. The TraPPE-UA force field for methanol is used [52].

Example 3: Binary Mixture CO₂ and CH₄

In this example, the phase composition of the CO₂-CH₄ binary mixture is calculated at 230 K in the Gibbs Ensemble at constant pressure (40 atm). The single-particle force field for CO₂ and CH₄ reported by Li and Calo is used [53].

8.3 Reaction Ensemble

Example 1: Reaction in an Ideal Gas

In this example a fictive reaction Urea \leftrightarrow Urea (dummy) is simulated for an color mixture of urea treated as an ideal gas. The activity coefficient of an ideal gas component is equal to 1. The lambda probability distribution for Urea should be flat. The temperature is 320 K and the pressure is 1 bar.

Example 2: Haber-Bosch Reaction

The ammonia synthesis reaction, N₂+3H₂ \leftrightarrow 2NH₃, is simulated at 573 K and 20 MPa. In this example, the starting composition is close to the equilibrium composition. The partition functions of the isolated molecule for N₂, H₂ and NH₃ at T=573 K, 673 K, 773 K and 873 K are reported in the Supporting Information of [1]. The simulation method is also explained in detail in this paper. Details on the computation of the partition function can be found in the Supporting Information of [6].

8.4 Grand-Canonical and Osmotic Ensemble

Example 1: Lennard-Jones particles in the Grand-Canonical Ensemble

In this example, a mixture of LJ particles (A+B) is simulated in the Grand-Canonical Ensemble at a reduced temperature of 2. The number of molecules of type B is fixed at 200. A fractional molecule of type A is used to compute the number of molecules of A that achieves the density of LJ particles at this temperature and fugacity coefficient of 1.9877. The number of molecules of type B resulting from this simulation is 200. This achieves a density of 0.765 (dimensionless unit). This is in agreement with simulations in the *NPT* ensemble for the same system, where the density equals 0.766.

Example 2: Lennard-Jones particles in the Osmotic Ensemble

A mixture of LJ particles (A and B) is simulated in the Osmotic Ensemble at a reduced temperature of 2. The number of molecules of type B is fixed at 200. A fractional molecule of type A is used to compute the number of molecules of A that achieves the density of LJ particles at this temperature and fugacity coefficient of 1.9877. In addition, volume change trial moves are used. The number of molecules of type B resulting from this simulation is 200. This achieves a density of 0.765 (dimensionless unit). This is in agreement with simulations in the *NPT* ensemble for the same system, where the density equals 0.766.

Example 3: Solubility of CO₂ in Ethylene glycol

In this example, the solubility of CO₂ in ethylene glycol is computed at 373 K and 5.39 bar. The Osmotic Ensemble is used, where a fractional molecule is used for the insertion and deletion of CO₂ molecules. The fugacity coefficient is computed using the PC-SAFT Equation of State [54]. The computed number of CO₂ molecules is 2.4. At the same conditions, experiments predict that 2.08 CO₂ molecules are absorbed.

8.5 Ionic Liquids

Example 1: Density of [bmim][Tf2N]

The density of [bmim][Tf2N] is computed at 333 K and 1 bar, using the *NPT* ensemble. The computed density is 1416 kg/m³, a value close to experimental measurements (1402 kg/m³). The CFC method is not used in this example.

Example 2: Solubility of CO₂ in [bmim][Tf2N] using the *NPT* ensemble

The excess chemical potential and the Henry constant of CO₂ are calculated in [bmim][Tf2N] at 333.15 K and 1 bar in the *NPT* ensemble. A fractional molecule of CO₂ is used and in addition to Lambda Trial move, the CFC hybrid move is used to improve sampling. The excess chemical potential and the Henry constant of CO₂ are computed to be -116 K and 6.64 MPa, respectively. The latter value agrees with experimental measurements (6.56 MPa).

Example 3: Solubility of CO₂ in [bmim][Tf2N] using the Grand-Canonical Ensemble

In this example, the solubility of CO₂ in [bmim][Tf2N] is computed at 333 K and 3.5 MPa. The Grand-Canonical Ensemble is used, combined with the CFC method. A fractional molecule of CO₂ facilitates insertion and deletion. The mole fraction of CO₂ (mole of CO₂ per mole of ion pairs) is computed to be 0.34, comparable to the experimental value (0.38).

Example 4: [bmim][Tf2N] as an Ideal Gas

In this example, [bmim][Tf2N] is treated as an ideal gas molecule in the *NPT* ensemble. Only trial moves that change the internal structure of the molecule are used. Figure 1 in the EXAMPLES shows the number of molecules-volume relationship at constant pressure (1 bar) and temperature

(298 K). In Figure 2 in the EXAMPLES, the volume is plotted as a function of pressure at constant temperature (298 K) and number of molecules (25 ion pairs). Both figures demonstrate how the simulation results agree with those computed analytically from the ideal gas law.

Chapter 9

Troubleshooting

In this section we discuss possible WARNING and ERROR messages. Warning messages are there to warn the user if some input for a simulation might be undesired. The simulation will still run and it is up to the user to decide whether the message can be ignored or not. Error messages always result in termination of the simulation because of various reasons. The input needs to be changed in order to run and finish a simulation properly.

9.1 Warning Messages

Vapor phase box has changed during the simulation

Appears at the end of a simulation if cluster trial moves have been used in a simulation and during the simulation the initial vapor box became the liquid and the liquid box became the vapor. This is important to notice because cluster trial moves should typically only be performed in the vapor box.

Read from ./INPUT/restart.in but something might be wrong with the format. Check the Warning/Error messages.

Appears after reading a restart file. It is possible that bondlengths and bondangles have changed because of rounding errors. This will also be indicated by other warning or error messages.

Sigma of Atom Type ... is very small.

Appears while reading the file `forcefield.in` if $\sigma < 0.1$. This might be a typo but can also be intentional.

Equilibration angle of Bending Type ... is not in the interval (-180,180).

Appears while reading the file `forcefield.in` if the equilibrium angle of some bending potential is not between -180° and 180° .

Default ... used:

Appears while reading the file `forcefield.in` if some force field parameters are unset. In that case the default values will be used which are reported in this warning message.

No correction for truncating Lennard-Jones interactions chosen.

Appears while reading the file `forcefield.in` if no method is chosen to take truncation effects into account. This might be intentional in which case this message can be ignored.

Scaling of intramolecular interactions (bending, torsion, LJ and EL) has been switched off for all Molecule Types in Fractional Groups of Type RxMC and GCMC.

Appears while reading the file `topology.in` if fractionals of type GCMC are defined or a reaction is defined that contains molecules where intramolecular interactions are scaled. In that case the scaling is switched off because this should not happen in CFCMC simulations.

dvolume is large compared to the Box size of Box ...

Appears while reading the file `topology.in` if the maximum ΔV for the volume change trial move is larger than 10% of the initial volume. Especially in the vapor phase this can lead to an error when the cutoff radius is larger than half the box size.

dlambda is negative in Box ... for Fractional ...

Appears while reading the file `topology.in` if the maximum $\Delta\lambda$ for the lambda trial move is negative.

dlambda is larger than 1 in Box ... for Fractional ...

Appears while reading the file `topology.in` if the maximum $\Delta\lambda$ for the lambda trial move is larger than 1. If the maximum change is larger then a lot of lambda trial moves will be rejected during the simulation simply because the probability that λ ends outside the interval $(0, 1)$ increases a lot.

LambdaSwitch is large for charged Fractional Group ...

Appears while reading the file `topology.in` if a fractional group has charged molecules and LambdaSwitch is larger than 0.9. This could make simulations less efficient because the available interval to build up electrostatic interactions is to short.

LambdaSwitch is large for charged Fractional Group ..., Reaction Step ... has a charge.

Similar to the message above but now for the reaction step in a fractional group.

No Fractionals of type ... so those moves have been switched off.

Appears while reading the file `settings.in` if CFCMC trial moves have been selected but no fractionals of that type have been defined.

9.2 Error Messages

(Intermolecular/Intramolecular) Energy Overlap (...)

This message indicates that the distance between two atoms is smaller than the minimum distance $r_{\min,ij}$. This should not happen during a simulation and is most likely to be a mistake in the source code. Note that the text between the parentheses indicates in which routine/trial move the overlap was first detected: the real overlap can be introduced already at a different place.

N_MolType <= 0

At least 1 Molecule Type should be defined for a simulation.

N_MolType > MaxMolType

N_Frac < 0 or N_Frac > MaxFrac

N_MolInFrac > MaxMolInFrac

N_Reaction < 0 or N_Reaction > MaxReaction

N_MolInReactionStep > MaxMolInFrac

N_MolTypePair < 0 or N_MolTypePair > MaxMolTypePair

N_AtomType < 0 or Natomtype > MaxAtomType

N_AtomInMolType <= 0 or N_AtomInMolType > MaxAtom. Molecule Type: ...

N_BondInMolType < 0 or N_BondInMolType > MaxBond. Molecule Type: ...

N_BendingInMolType < 0 or N_BendingInMolType > MaxBendingInMolType. Molecule Type: ...

N_TorsionInMolType < 0 or N_TorsionInMolType > MaxTorsionInMolType. Molecule Type: ...

Kmax for Ewald is larger than MaxKvec in Box ...

These messages indicate either that a value is less than 0 or larger than the maximum declared dimension in the file `maximum_dimensions.inc`. Increase the value of the Parameter in `maximum_dimensions.inc` and **recompile**.

Molecule file not present in INPUT folder: ...

Each molecule that appears in the file `topology.in` should be in the INPUT directory with exactly the same name.

... Type not defined in force field: ...

An Atom/Bending/Torsion Type that is used in one of the Molecule Types cannot be found in the file `forcefield.in`, check if the Type is defined and the label/name is exactly the same.

Number of ... Types is wrong.

This indicates an error in reading the file `forcefield.in`. Most times this is caused by a wrong number on the first lines.

Torsion Type not known. Torsion type: ...

Appears if a Torsion Type is unknown (only possibilities are TraPPE, OPLS and Ryckaert-Bellemans).

No method for calculating Electrostatic Interactions chosen or method unknown.

Appears if a no method for calculating electrostatic interactions is chosen (only possibilities are Ewald, Wolf, Fennell-Gezelter and none).

No Mixing Rules for Lennard-Jones chosen or Mixing Rules unknown.

Appears if a no Lennard-Jones mixing rules are chosen (only possibilities are Lorentz-Berthelot and Jorgensen).

Number of ... is wrong in molecule ...

This indicates an error in reading molecule definition files. Most times this is caused by a wrong

number on the first lines.

Fractional of Type GE defined but only 1 Simulation Box defined.

Appears if N_Box is set to 1 in the file `settings.in` and a Fractional Group of Type GE is defined.

Number of Molecules in Fractional of Type GC should be 1. Please check Fractional

...

The GCMC works for Fractional Groups that contain only one molecule.

Fractional Type of Fractional number ... not defined: ...

Appears if a Fractional Type is defined but unknown (only possibilities are: NVT, NPT, GE and GCMC).

Box ... is not charge neutral.

Appears if the sum of charges in a simulation box is not zero. Calculation of electrostatic interactions is only possible if the system does not have a net charge. Change the number of molecules such that the total charge equals zero.

Fractional ... is not charge neutral.

Appears if the sum of charges in a Fractional Group is not zero. Fractionals that have a net charge introduce a net charge to the system which interferes with the calculation of electrostatic interactions. Change the Fractional Group such that the charge equals zero.

Reaction Step ... of Reaction ... is not charge neutral.

Similar to the message above but now for a reaction step.

Volume; Rcut is too large in Box ...

Boxsize not large enough for Rcut LJ in box: ...

Boxsize not large enough for Rcut Electrostatic in box: ...

The cutoff radius should not exceed half the Box size. Increase the Box size or decrease the cutoff radii. It is important to note that the value of the cutoff radius is part of the force field, so changing the cut off radius will change all thermodynamic properties. One can also get this error if the number of molecules is too small, or when one attempts to perform the NPT version of the Gibbs ensemble for a pure component system (as this violates the phase rule).

Check bond, attempt to check for bond between the same atom.

Appears if the topology of a molecule is read and a bond is set between twice the same atom.

Sum of number of molecules in boxes is not equal to total number of molecules

Sum of number of molecules and fractionals is not equal to total number of molecules

Difference in number of whole molecules per type

Molecule is more than one time in a list ...

Molecule is not in list of molecules ...

Molecule in list is of wrong type

Molecule in list is in wrong box

Value of lambda

Component of fractional not in right box

Molecule is not labeled as fractional ...

Wrong center of mass for molecule ...

Bond ... in molecule ... has changed length

These messages appear when the system is checked for inconsistencies. When such a message appears it most likely indicated a programming error.

Chapter 10

References

- [1] Poursaeidesfahani, A.; Hens, R.; Rahbari, A.; Ramdin, M.; Dubbeldam, D.; Vlugt, T. J. H. *Journal of Chemical Theory and Computation* **2017**, *13*, 4452-4466.
- [2] Rahbari, A.; Hens, R.; Nikolaidis, I. K.; Poursaeidesfahani, A.; Ramdin, M.; Economou, I. G.; Moultsos, O. A.; Dubbeldam, D.; Vlugt, T. J. H. *Molecular Physics* **2018**, *116*, 3331-3344.
- [3] Rahbari, A.; Brenkman, J.; Hens, R.; Ramdin, M.; van den Broeke, L. J. P.; Schoon, R.; Henkes, R.; Moultsos, O. A.; Vlugt, T. J. H. *Journal of Chemical & Engineering Data* **2019**, *64*, 4103-4115.
- [4] Caro-Ortiz, S.; Hens, R.; Zuidema, E.; Rigutto, M.; Dubbeldam, D.; Vlugt, T. J. H. *Fluid Phase Equilibria* **2019**, *485*, 239 - 247.
- [5] Caro-Ortiz, S.; Hens, R.; Zuidema, E.; Rigutto, M.; Dubbeldam, D.; Vlugt, T. J. H. *Fluid Phase Equilibria* **2020**, *506*, 112370.
- [6] Hens, R.; Rahbari, A.; Caro-Ortiz, S.; Dawass, N.; Erdős, M.; Poursaeidesfahani, A.; Salehi, H. S.; Celebi, A. T.; Ramdin, M.; Moultsos, O. A.; Dubbeldam, D.; Vlugt, T. J. H. *Journal of Chemical Information and Modeling* **2020**, *60*, 2678-2682.
- [7] Polat, H. M.; Salehi, H. S.; Hens, R.; Wasik, D. O.; de Meyer, F.; Houriez, C.; Coquelet, C.; Calero, S.; Dubbeldam, D.; Moultsos, O. A.; Vlugt, T. J. H. *Journal of Chemical Information and Modeling* **2021**, *61*, 3752-3757.
- [8] Rahbari, A.; Hens, R.; Ramdin, M.; Moultsos, O. A.; Dubbeldam, D.; Vlugt, T. J. H. *Molecular Simulation* **2021**, *47*, 804-823.
- [9] Hens, R. *Molecular Simulation of Phase and Reaction Equilibria: Software and Algorithm Development*, Thesis, Delft University of Technology, 2020 <https://homepage.tudelft.nl/v9k6y/thesis-Hens.pdf>.
- [10] Rahbari, A. *Thermodynamics of industrially relevant systems: method development and applications*, Thesis, Delft University of Technology, 2020 <https://homepage.tudelft.nl/v9k6y/thesis-Rahbari.pdf>.

- [11] “Intel Fortran Compiler”, <https://software.intel.com/en-us/fortran-compilers>.
- [12] “GNU Fortran”, <https://gcc.gnu.org/fortran/>.
- [13] Frenkel, D.; Smit, B. *Understanding molecular simulation: from algorithms to applications*; Academic Press: San Diego, California, 2 ed.; 2002.
- [14] Matsumoto, M.; Nishimura, T. *ACM Transactions on Modeling and Computer Simulation* **1998**, *8*, 3–30.
- [15] Widom, B. *The Journal of Chemical Physics* **1963**, *39*, 2808-2812.
- [16] Sindzingre, P.; Cicotti, G.; Massobrio, C.; Frenkel, D. *Chemical Physics Letters* **1987**, *136*, 35 - 41.
- [17] Wang, F.; Landau, D. P. *Physical Review Letters* **2001**, *86*, 2050–2053.
- [18] Martin, M. G.; Siepmann, J. I. *The Journal of Physical Chemistry B* **1998**, *102*, 2569-2577.
- [19] Jorgensen, W. L.; Madura, J. D.; Swenson, C. J. *Journal of the American Chemical Society* **1984**, *106*, 6638-6646.
- [20] Ryckaert, J.-P.; Bellemans, A. *Chemical Physics Letters* **1975**, *30*, 123 - 125.
- [21] Wolf, D.; Kebinski, P.; Phillipot, S. R.; Eggebrecht, J. *The Journal of Chemical Physics* **1999**, *110*, 8254-8282.
- [22] Fennell, C. J.; Gezelter, J. D. *The Journal of Chemical Physics* **2006**, *124*, 234104.
- [23] Ewald, P. P. *Annalen der Physik* **1921**, *369*, 253–287.
- [24] Stubbs, J. M.; Potoff, J. J.; Siepmann, J. I. *The Journal of Physical Chemistry B* **2004**, *108*, 17596-17605.
- [25] Rahbari, A.; Hens, R.; Moulton, O. A.; Dubbeldam, D.; Vlugt, T. J. H. *Journal of Chemical Theory and Computation* **2020**, *16*, 1757-1767.
- [26] Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Great Clarendon Street, Oxford, OX2 6DP, United Kingdom, Second ed.; 2017.
- [27] Jorgensen, W. L.; Tirado-Rives, J. *Journal of the American Chemical Society* **1988**, *110*, 1657-1666.
- [28] Anwar, J.; Heyes, D. M. *The Journal of Chemical Physics* **2005**, *122*, 224117.
- [29] Shi, W.; Maginn, E. J. *Journal of Computational Chemistry* **2008**, *29*, 2520–2530.
- [30] Vlugt, T. J. H.; Garcia-Perez, E.; Dubbeldam, D.; Ban, S.; Calero, S. *Journal of Chemical Theory and Computation* **2008**, *4*, 1107-1118.
- [31] Dubbeldam, D.; Walton, K.; Vlugt, T. J. H.; Calero, S. *Adv. Theory Simul.* **2019**, *2*, 190035.
- [32] Hens, R.; Vlugt, T. J. H. *Journal of Chemical & Engineering Data* **2018**, *63*, 1096-1102.

- [33] Rahbari, A.; Hens, R.; Jamali, S. H.; Ramdin, M.; Dubbeldam, D.; Vlugt, T. J. H. *Molecular Simulation* **2019**, *45*, 336-350.
- [34] Rahbari, A.; Hens, R.; Dubbeldam, D.; Vlugt, T. J. H. *Molecular Physics* **2019**, *117*, 3493-3508.
- [35] Gonzalez-Salgado, D.; Vega, C. *The Journal of Chemical Physics* **2016**, *145*, 034508.
- [36] “Alchemlyb”, <https://alchemlyb.readthedocs.io/en/latest/>.
- [37] Panagiotopoulos, A.; Quirke, N.; Stapleton, M.; Tildesley, D. *Molecular Physics* **1988**, *63*, 527-545.
- [38] Poursaeidesfahani, A.; Torres-Knoop, A.; Dubbeldam, D.; Vlugt, T. J. H. *Journal of Chemical Theory and Computation* **2016**, *12*, 1481-1490.
- [39] Johnson, J. K.; Panagiotopoulos, A. Z.; Gubbins, K. E. *Molecular Physics* **1994**, *81*, 717-733.
- [40] Smith, W. R.; Triska, B. *The Journal of Chemical Physics* **1994**, *100*, 3019-3027.
- [41] Adams, D. *Molecular Physics* **1975**, *29*, 307-311.
- [42] Shi, W.; Maginn, E. J. *Journal of Chemical Theory and Computation* **2007**, *3*, 1451-1463.
- [43] Orkoulas, G.; Panagiotopoulos, A. Z. *The Journal of Chemical Physics* **1994**, *101*, 1452-1459.
- [44] Rossky, P.; Doll, J.; Friedman, H. *The Journal of Chemical Physics* **1978**, *69*, 4628-4633.
- [45] Duane, S.; Kennedy, A.; Pendleton, B. J.; Roweth, D. *Physics Letters B* **1987**, *195*, 216-222.
- [46] Verlet, L. *Physical Review* **1967**, *159*, 98-103.
- [47] Hairer, E.; Lubich, C.; Wanner, G. *Acta Numerica* **2003**, *12*, 399-450.
- [48] Miller, T. F.; Eleftheriou, M.; Pattnaik, P.; Ndirango, A.; Newns, D.; Martyna, G. J. *Journal of Chemical Physics* **2002**, *116*, 8649-8659.
- [49] Todorov, I. T.; Smith, W.; Trachenko, K.; Dove, M. T. *J. Mater. Chem.* **2006**, *16*, 1911-1918.
- [50] Kamberaj, H.; Low, R. J.; Neal, M. P. *Journal of Chemical Physics* **2005**, *122*, 224114.
- [51] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *The Journal of Chemical Physics* **1983**, *79*, 926-935.
- [52] Chen, B.; Potoff, J. J.; Siepmann, J. I. *The Journal of Physical Chemistry B* **2001**, *105*, 3093-3104.
- [53] Li, J.; Calo, V. M. *Journal of Computational Physics* **2013**, *249*, 233 - 248.
- [54] Gross, J.; Sadowski, G. *Industrial & Engineering Chemistry Research* **2001**, *40*, 1244-1260.