

---

# **GOMC Documentation**

***Release 2.75***

**GOMC Development Group**

**May 02, 2022**



## CONTENTS:

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Release 2.75 Notes</b>	<b>3</b>
<b>3</b>	<b>Introduction</b>	<b>5</b>
3.1	GOMC supported ensembles: . . . . .	5
3.2	GOMC supported Monte Carlo moves: . . . . .	5
3.3	GOMC supported force fields: . . . . .	6
3.4	GOMC supported molecules: . . . . .	6
<b>4</b>	<b>Software Requirements</b>	<b>9</b>
4.1	C++11 Compliant Compiler . . . . .	9
4.2	CMake . . . . .	10
4.3	CUDA Toolkit . . . . .	11
4.4	MPI (Optional for Standard; Required for MultiSim) . . . . .	11
<b>5</b>	<b>Recommended Software Tools</b>	<b>13</b>
5.1	Packmol . . . . .	13
5.2	VMD . . . . .	13
5.3	Molecular Simulation Design Framework (MoSDeF) . . . . .	15
<b>6</b>	<b>How to get the software</b>	<b>17</b>
6.1	GitHub . . . . .	17
6.2	Website . . . . .	18
<b>7</b>	<b>Compiling GOMC</b>	<b>21</b>
7.1	Linux . . . . .	21
7.2	Windows . . . . .	22
7.3	Configuring CMake . . . . .	22
<b>8</b>	<b>GPU-Accelerated GOMC</b>	<b>25</b>
<b>9</b>	<b>Input File Formats</b>	<b>27</b>
9.1	PDB File . . . . .	27
9.2	PSF File . . . . .	30
9.3	Topology File . . . . .	32
9.4	Parameter File . . . . .	33
9.5	Exotic or Mie Parameter File . . . . .	39
9.6	Control File (*.conf) . . . . .	41
9.7	XSC (eXtended System Configuration file) File . . . . .	47
9.8	COOR (binary coordinates) File . . . . .	47

9.9	VEL (binary velocity) File . . . . .	47
9.10	CHK (checkpoint) File . . . . .	47
<b>10</b>	<b>GOMC's Output Files</b>	<b>69</b>
10.1	Console Output . . . . .	69
10.2	Block Output Files . . . . .	71
10.3	PDB Output Files . . . . .	77
10.4	PSF Output File . . . . .	80
10.5	Molecule Distribution Output File . . . . .	82
10.6	Histogram Output File . . . . .	82
10.7	Free Energy Output File . . . . .	85
<b>11</b>	<b>Putting it all together: Running a GOMC Simulation</b>	<b>87</b>
<b>12</b>	<b>Intermolecular Energy and Virial Function (Van der Waals)</b>	<b>91</b>
12.1	VDW . . . . .	91
12.2	EXP6 . . . . .	92
12.3	SHIFT . . . . .	92
12.4	SWITCH . . . . .	94
12.5	SWITCH (MARTINI) . . . . .	95
<b>13</b>	<b>Intermolecular Energy and Virial Function (Electrostatic)</b>	<b>99</b>
13.1	Ewald . . . . .	99
13.2	VDW . . . . .	100
13.3	EXP6 . . . . .	101
13.4	SHIFT . . . . .	101
13.5	SWITCH . . . . .	101
13.6	SWITCH (MARTINI) . . . . .	102
<b>14</b>	<b>Long-range Correction (Energy and Virial)</b>	<b>103</b>
14.1	VDW . . . . .	103
14.2	EXP6 . . . . .	104
<b>15</b>	<b>Coupling Interaction with <math>\lambda</math></b>	<b>105</b>
15.1	VDW . . . . .	105
15.2	Electrostatic . . . . .	105
15.3	Long-range Correction (VDW) . . . . .	106
15.4	Long-range Correction (Electrostatic) . . . . .	107
<b>16</b>	<b>Hybrid Monte Carlo-Molecular Dynamics (MCMD)</b>	<b>109</b>
16.1	GOMC Requirements . . . . .	109
16.2	NAMD Requirements . . . . .	109
16.3	Dynamic Subvolumes for Dual Control Volume Molecular Dynamics . . . . .	110
16.4	Run a Hybrid Monte Carlo-Molecular Dynamics Sim . . . . .	111
<b>17</b>	<b>How to?</b>	<b>113</b>
17.1	Visualizing Simulation . . . . .	113
17.2	Build molecule and topology file . . . . .	114
17.3	Simulate rigid molecule . . . . .	116
17.4	Restart the simulation . . . . .	116
17.5	Recalculate the energy . . . . .	119
17.6	Simulate adsorption . . . . .	120
17.7	Calculate Solvation Free Energy . . . . .	124
17.8	Run a Multi-Sim . . . . .	126





## **OVERVIEW**

This document will instruct a new user how to download, compile, prepare the input files, and run the GOMC molecular simulation code. A basic understanding of statistical physics is recommended to complete this tutorial.

To demonstrate the capabilities of the code, the user is guided through the process of downloading, compiling a GOMC executable, and preparing input files such as PDB, PSF, Parameter, and Configuration file. Executable is then used to calculate the saturated vapor and liquid equilibria (VLE) using Gibbs Ensemble Monte Carlo on systems of pure isobutane (R600a), a branched alkane that whose application as a refrigerant/propellant is increasing. The Transferable Potentials for Phase Equilibria (TraPPE) united atom (UA) force field is used to describe the molecular geometry constraints and the intermolecular interactions.

<http://en.wikipedia.org/wiki/Isobutane>





## RELEASE 2.75 NOTES

Certain changes have been made which differ from previous GOMC behavior. New features have been added to assist users and developers in compiling, running, and analyzing, improve reproducibility, increase the capacity of GOMC to simulate biological molecules, perform Hybrid Monte-Carlo/Molecular Dynamics simulations, and increase performance. A non-comprehensive list is provided below.

### Differing behavior:

Previous GOMC versions used REMARKS in the PDB header to save box dimensions and random number generator state. While this is still currently partially supported, it is in the process of being deprecated and checkpointing should be used. Secondly, restarting GOMC should no longer be performed using the merged files containing both boxes (\*.BOX\_0.pdb, \*.merged.psf) which are produced solely for visualization. Furthermore, The user should now use the box-specific restart files (pdb, psf, xsc, coor, chk) as input. Finally, the pdb trajectory files (\*.BOX\_0.pdb) are in the process of being deprecated and replaced by binary trajectory files (\*.BOX\_0.dcd), though both are currently available.

### Updated Manual Sections:

#### (1) Introduction

##### GOMC supported Monte Carlo moves

Force-biased Multiparticle move (Rigid-body displacement or rotation of all molecules) Brownian Motion Multiparticle move (Rigid-body displacement or rotation of all molecule Non-Equilibrium Molecule Transfer Inter-box subvolume targeted swap Intra-box subvolume targeted swap

##### GOMC supported molecules

Biological molecules which consist of multiple residues are now supported. Care should be taken when generating the molecules such that all bonds, angles, and dihedrals are included in the PSF file. Support for these molecules is experimental.

#### (2) Recommended Software Tools

Molecular Simulation Design Framework (MoSDeF)

#### (3) Compiling GOMC

./metamake.sh [OPTIONS] [ARGUMENTS]

##### OPTIONS

<b>-d</b>	Compile in debug mode.
<b>-g</b>	Compile with gcc.
<b>-m</b>	Compile with MPI enabled.
<b>-p</b>	Compile with NVTX profiling for CUDA
<b>-t</b>	Compile Google tests.

(4) GPU-accelerated GOMC

A section describing the GPU-accelerated regions of GOMC code is included.

(5) Input File Formats

Support for binary coordinates, trajectories, box dimensions, velocities, and checkpoint files are included. Checkpoint files guarantee trajectory files can be concatenated, along with ensuring no deviation from a single simulation's results in a simulation which was interrupted and restarted.

(5a) Restart a simulation from a checkpoint

(5b) Restart from binary coordinates and box dimensions

(5c) Target insertions to subvolumes

(5d) Overwrite start step

(6) Hybrid Monte Carlo-Molecular Dynamics (MCMD)

Instructions on running an alternating Hybrid MCMD algorithm using GOMC and NAMD are included.

## INTRODUCTION

GPU Optimized Monte Carlo (GOMC) is open-source software for simulating many-body molecular systems using the Metropolis Monte Carlo algorithm. GOMC is written in object oriented C++, which was chosen since it offers a good balance between code development time, interoperability with existing software elements, and code performance. The software may be compiled as a single-threaded application, a multi-threaded application using OpenMP, or to use many-core heterogeneous CPU-GPU architectures using OpenMP and CUDA. GOMC officially supports Windows 7 or newer and most modern distribution of GNU/Linux. This software has the ability to compile on recent versions of macOS; however, such a platform is not officially supported.

GOMC employs widely-used simulation file types (PDB, PSF, CHARMM-style parameter file) and supports polar and non-polar linear and branched molecules. GOMC can be used to study vapor-liquid and liquid-liquid equilibria, adsorption in porous materials, surfactant self-assembly, and condensed phase structure for complex molecules.

To cite GOMC software, please refer to [GOMC paper](#).

### 3.1 GOMC supported ensembles:

- Canonical (NVT)
- Isobaric-isothermal (NPT)
- Grand canonical ( $\mu$  VT)
- Constant volume Gibbs (NVT-Gibbs)
- Constant pressure Gibbs (NPT-Gibbs)

### 3.2 GOMC supported Monte Carlo moves:

- Rigid-body displacement
- Rigid-body rotation
- [Force-biased Multiparticle](#) move (Rigid-body displacement or rotation of all molecules)
- [Brownian Motion Multiparticle](#) move (Rigid-body displacement or rotation of all molecules)
- Regrowth using [coupled-decoupled configurational-bias](#)
- Crankshaft using combination of [crankshaft](#) and [coupled-decoupled configurational-bias](#)
- Intra-box swap using [coupled-decoupled configurational-bias](#)
- Intra-box [molecular exchange Monte Carlo](#)
- Intra-box targeted swap using [coupled-decoupled configurational-bias](#)

- Inter-box swap using `coupled-decoupled configurational-bias`
- Inter-box targeted swap using `coupled-decoupled configurational-bias`
- Inter-box `molecular exchange monter carlo`
- Non-Equilibrium Molecule Transfer <<https://journals.aps.org/pre/abstract/10.1103/PhysRevE.66.046705>>`\_\_
- Volume exchange (both isotropic and anisotropic)

### 3.3 GOMC supported force fields:

- OPLS
- CHARMM
- TraPPE
- Mie
- Martini

### 3.4 GOMC supported molecules:

- Polar molecules (using Ewald summation)
- Non-polar molecules (standard LJ and Mie potential)
- Linear molecules (using `coupled-decoupled configurational-bias`)
- Branched molecules (using `coupled-decoupled configurational-bias`)
- Cyclic molecules (using combination of `coupled-decoupled configurational-bias` and `crankshaft` to sample intramolecular degrees of freedom of cyclic molecules)
- Large biomolecules can be loaded into GOMC (although current sampling is limited to `crankshaft` to sample intramolecular degrees of freedom)

---

**Note:**

- Biomolecules often have defined secondary structure which is maintained through improper terms, CMAP, and missing angles and dihedrals.
  - These complexities make sampling incorrect (improper, CMAP) or impossible (missing angles and dihedrals) in GOMC and these molecules should be held fixed.
- 

---

**Note:**

- It is important to start the simulation with correct molecular geometry such as correct bond length, angles, and dihedral.
  - In GOMC if the defined bond length in `Parameter` file is different from calculated bond length in PDB files by more than 0.02 , you will receive a warning message with detailed information (box, residue id, specified bond length, and calculated bond length)
- 

---

**Important:**

- Molecular geometry of `Linear` and `Branched` molecules will be corrected during the simulation by using the Monte Carlo moves that uses coupled-decoupled configurational-bias method, such as `Regrowth`, `Intra-box swap`, and `Inter-box swap`.
- 

**Warning:**

- Bond length of the `Cyclic` molecules that belong to the body of rings will never be changed. Incorrect bond length may result in incorrect simulation results.
- To sample the angles and dihedrals of a `Cyclic` molecule that belongs to the body of the ring, `Regrowth` or `Crankshaft` Monte Carlo move must be used.
- Any atom or group attached to the body of the ring, will uses coupled-decoupled configurational-bias to sample the molecular geometry.
- Flexible `Cyclic` molecules with multiple rings (3 or more) that share edges (e.g. tricyclic), are not supported in GOMC. This is due the fact that no `Crankshaft` move can alter the angle or dihedral of this atom, without changing the bond length.



## SOFTWARE REQUIREMENTS

### 4.1 C++11 Compliant Compiler

- Linux/macOS

- icpc (Intel C++ Compiler)

In Linux, the Intel compiler will generally produce the fastest CPU executables (when running on Intel Core processors). Type the following command in a terminal:

```
$ icpc --version
```

If gives a version number 16.0.3 (2016 Initial version) or later, you're all set. Otherwise, we recommend upgrading.

- g++

Type the following command in a terminal:

```
$ g++ --version
```

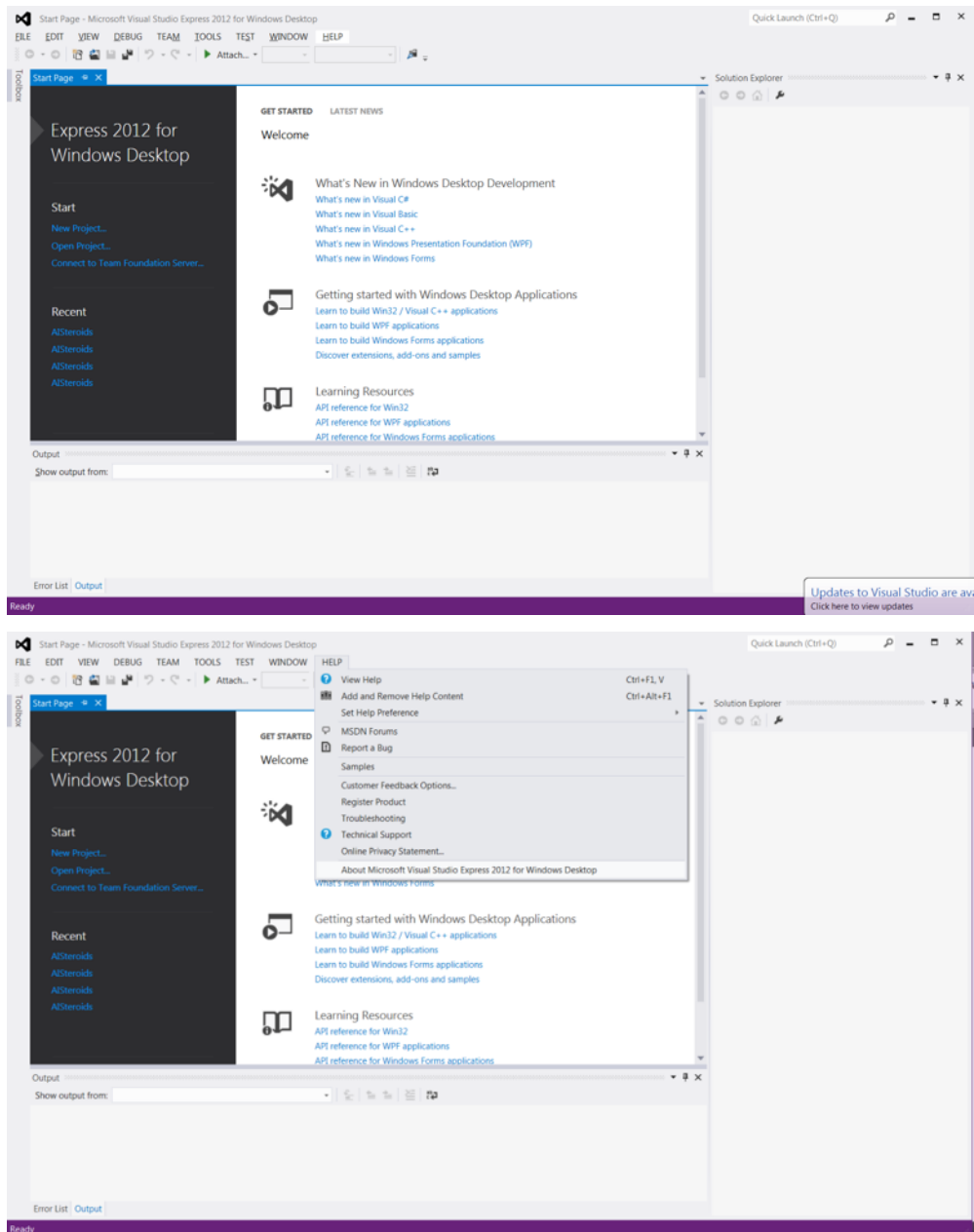
If gives a version number 4.4 or later, you're all set. Otherwise, we recommend upgrading.

- Windows

Visual Studio Microsoft's Visual Studio 2010 or later is recommended.

To check the version:

*Help (top tab) -> About Microsoft Visual Studio*



## 4.2 CMake

To check if cmake is installed:

```
$ which cmake
```

To check the version number:

```
$ cmake --version
```

The minimum required version is 2.8. However, we recommend to use version 3.2 or later.



## 4.3 CUDA Toolkit

CUDA is required to compile the GPU executable in both Windows and Linux. However, is not required to compile the CPU code. To download and install CUDA visit NVIDIA's webpage:

<https://developer.nvidia.com/cuda-downloads>

<https://developer.nvidia.com/cuda>

Please refer to CUDA Developer webpages to select an appropriate version for the desired platform. To install CUDA in Linux root/sudo, privileges are generally required. In Windows, administrative access is required.

To check if nvcc is installed:

```
$ which nvcc
```

To check the version number:

```
$ nvcc --version
```

The GPU builds of the code requires NVIDIA's CUDA 8.0 or newer.

## 4.4 MPI (Optional for Standard; Required for MultiSim)

An MPI Library is required to compile the MPI version of GOMC in both Windows and Linux. However, it is not required to compile standard GOMC. There are a couple of options to install an MPI library.

- 1) We recommend the Intel MPI Library:

<https://software.intel.com/en-us/mpi-library>

- 2) The alternative we recommend to Intel MPI is MPICH. MPICH binary packages are available in many UNIX distributions and for Windows. For example, you can search for it using "yum" (on Fedora), "apt" (Debian/Ubuntu), "pkg\_add" (FreeBSD) or "port"/"brew" (Mac OS).

```
$ sudo apt-get install mpich
```

- 3) Another option is the OpenMPI library.

```
$ sudo apt-get install openmpi-bin openmpi-common openssh-client openssh-
↪server libopenmpi2 libopenmpi-dev
```



## RECOMMENDED SOFTWARE TOOLS

The listed programs are used in this manual and are generally considered necessary.

### 5.1 Packmol

Packmol is a free molecule packing tool (written in Fortran), created by José Mario Martínez, a professor of mathematics at the State University of Campinas, Brazil. Packmol allows a specified number of molecules to be packed at defined separating distances within a certain region of space. More information regarding downloading and installing Packmol is available on their homepage:

<http://www.ime.unicamp.br/~martinez/packmol>

**Warning:** One of Packmol's limitations is that it is unaware of topology; it treats each molecule or group of molecules as a rigid set of points. It is highly suggested to use the optimized structure of the molecule as the input file to packmol.

**Warning:** Another more serious limitation is that it is not aware of periodic boundary conditions (PBC). As a result, when using Packmol to pack PDBs for GOMC, it is recommended to pack to a box 1 Angstroms smaller than the simulation box size. This prevents hard overlaps over the periodic boundary.

### 5.2 VMD

VMD (Visual Molecular Dynamics) is a 3-D visualization and manipulation engine for molecular systems written in C-language. VMD is distributed and maintained by the University of Illinois at Urbana-Champaign. Its sources and binaries are free to download. It comes with a robust scripting engine, which is capable of running python and tcl scripts. More info can be found here:

<http://www.ks.uiuc.edu/Research/vmd/>

Although GOMC uses the same fundamental file types, PDB (coordinates) and PSF (topology) as VMD, it uses some special tricks to obey certain rules of those file formats. One useful purpose of VMD is visualization and analyze your systems.

Nonetheless, the most critical part of VMD is a tool called PSFGen. PSFGen uses a tcl or python script to generate a PDB and PSF file for a system of one or more molecules. It is, perhaps, the most convenient way to generate a compliant PSF file.

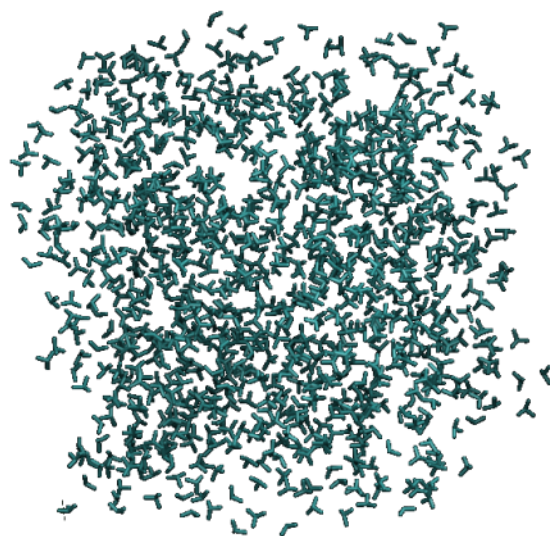


Fig. 1: A system of united atom isobutane molecules

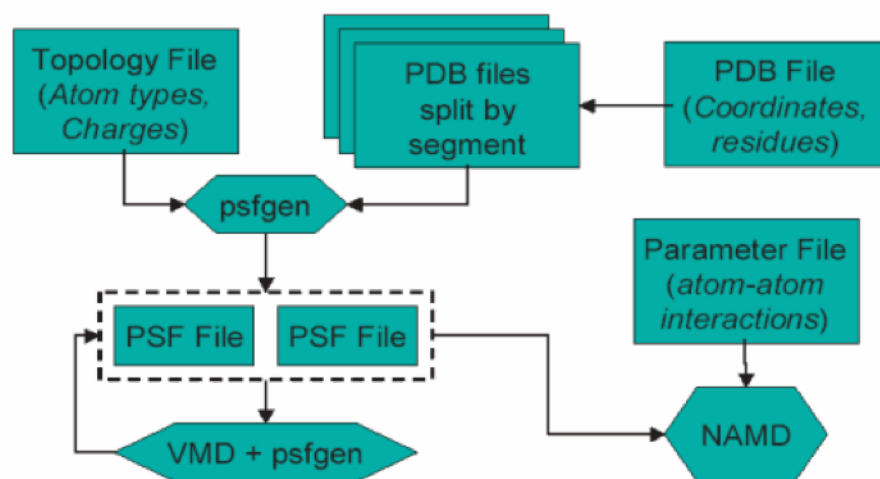


Fig. 2: An overview of the PSFGen file generation process and its relationship to VMD/NAMD

**Tip:** To read more about PSFGen, reference:

Plugin homepage @ UIUC

Generating a Protein Structure File (PSF), part of the NAMD Tutorial from UIUC

In-Depth Overview [PDF]

---

## 5.3 Molecular Simulation Design Framework (MoSDeF)

In this section, the MosDef python interface for creating customized GOMC simulations are discussed.

Link to documentation: [https://mbuild.mosdef.org/en/stable/getting\\_started/writers/GOMC\\_file\\_writers.html](https://mbuild.mosdef.org/en/stable/getting_started/writers/GOMC_file_writers.html)

Link to MosDef Examples Repository: <https://github.com/GOMC-WSU/GOMC-MoSDeF>

Link to Youtube tutorials : <https://www.youtube.com/playlist?list=PLdxD0z6HRx8Y9VhwcODxAHNQBBJDRvxMf>

Link to signac documentation: <https://signac.io/>

The Molecular Simulation Design Framework (MosDeF) is a GOMC-compatible software that allows these simulations to be transparent and reproducible and permits the easy generation of all the required files to run a GOMC simulation (the forcefield, coordinate, and topology files GOMC control file). The MoSDeF software also lowers the entry barrier for new users, minimizes the expert knowledge traditionally required to set up a simulation, and streamlines this process for more experienced users. MoSDeF is comprised of several conda packages (mBuild, foyer, and gmso), which are stand-alone packages; however, they are designed to all work seamlessly together, which is the case with MoSDeF-GOMC.

In general, molecules imported or built using mBuild, packed into a simulation box(s) and passed into the charmm writer function with additional parameters as arguments. The charmm writer then atom-types the simulation box's molecules using foyer, obtaining the molecular force field parameters. The next step utilizes the charmm writer to output the forcefield files, PDB/PSF, and GOMC control files, all the files needed to run a GOMC simulation. This MoSDeF-GOMC software is fully scriptable and compatible with Signac, allowing a fully automated and reproducible workflow.



## HOW TO GET THE SOFTWARE

The CPU and GPU code are merged together under GOMC project. Currently, version control is handled through the GitHub repository. The latest GOMC release, Example files, and User Manual can be downloaded from GOMC website or GitHub repository.

### 6.1 GitHub

The posted builds in Master branch are “frozen” versions of the code that have been validated for a number of systems and ensembles. Other branches are created as a means of implementing new features. The latest updated code builds, manual, example files, and other resources can be obtained via the following GitHub repository:

GOMC GitHub Repository

The screenshot shows the GitHub profile page for the GOMC repository. The header includes a search bar and navigation links: Pull requests, Issues, Marketplace, and Explore. The profile section on the left features the GOMC logo (a 3D cube with colored faces) and the text 'GOMC GOMC-WSU'. Below this is an 'Unfollow' button and contact information for Wayne State University, including the email 'gomc@eng.wayne.edu' and the website 'http://gomc.eng.wayne.edu'. The main content area, titled 'Overview', shows statistics: 6 Repositories, 0 Projects, 0 Stars, 22 Followers, and 0 Following. A 'Popular repositories' section lists several repositories: 'GOMC' (C++ code for parallel molecular simulation), 'GOMC\_Examples' (example files), 'Workshop' (instructions for learning about GOMC), 'Forcefields' (topologies and parameters), 'Benchmarks' (used in the GOMC paper), and 'Manual' (the GOMC Manual). Each repository entry includes its name, description, programming language, star count, and fork count.

GOMC and Examples repository can be found under the main page. Under GOMC repository, the code and manual can be found. Each repository can be downloaded by clicking on the Clone or download tab.

GOMC-WSU / GOMC

Unwatch 18 Unstar 19 Fork 14

Code Issues 6 Pull requests 2 Projects 0 Wiki Insights

GOMC - GPU Optimized Monte Carlo is a parallel molecular simulation code designed for high-performance simulation of large systems  
<http://gomc.eng.wayne.edu/>

monte-carlo monte-carlo-simulation gpu openmp adsorption free-energy phase-equilibrium surface-tension cuda  
 gibbs-ensemble grand-canonical-monte-carlo

462 commits 10 branches 12 releases 6 contributors AGPL-3.0

Branch: master New pull request Create new file Upload files Find File Clone or download

Clone with HTTPS ? Use Git or checkout with SVN using the web URL.  
<https://github.com/GOMC-WSU/GOMC.git>

Open in Desktop Download ZIP

.github/ISSUE_TEMPLATE	Update bug_report.md	
CMake	Support for Linear Molecule with atom order. Support fo	
lib	fixing version number and log file for new release	
src	fixed compiling issues on macos	
.gitignore	fixed gitignore	a year ago
.travis.yml	Added auto compiling; travisci	2 years ago
CMakeLists.txt	fixed compiling issues on macos	7 months ago
Change log	fixing version number and log file for new release	a year ago
GOMC_Config.h.in	GOMC Serial BETA 0.97	4 years ago
GOMC_Manual.pdf	Update manual	a year ago
License.txt	Change in License	2 years ago

To clone the GOMC using git, execute the following command in your terminal:

```
$ git clone https://github.com/GOMC-WSU/GOMC.git
```

To clone the GOMC Example files using git, execute the following command in your terminal:

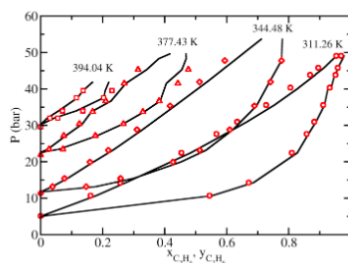
```
$ git clone https://github.com/GOMC-WSU/GOMC_Examples.git
```

## 6.2 Website

To access the GOMC website, please click on the following link: [GOMC Website](http://gomc.eng.wayne.edu/)

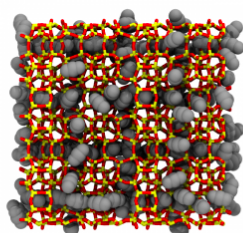
The code can be found under the download tab, below and to the right of the logo. When new betas (or release builds) are announced, they will replace the prior code under the downloads tab. An announcement will be posted on the front page to notify users.





### Vapor-Liquid Equilibrium

GOMC is capable of performing Gibbs ensemble Monte Carlo and grand canonical histogram-reweighting Monte Carlo simulations to predict the vapor-liquid equilibria of pure components and multicomponent mixtures. A variety of advanced configurational-bias algorithms, such as coupled-decoupled configurational-bias, molecular exchange Monte Carlo, configurational-bias regrowth, and crankshaft, are included to enhance the sampling of phase space.



### Adsorption

GOMC supports simulations of adsorption in rigid porous materials and can be used for high throughput screening of materials for gas storage and separation. A tool to take structures from the CoRE database and automatically setup simulations for high throughput screening may be found in the GitHub repository: <https://github.com/GOMC-WSU/Workshop/tree/HTS>. In the future, our HTS code will be integrated into the Molecular Simulation Design Framework (MoSDeF) toolkit.

GOMC is distributed as a compressed folder, containing the source and build system. To compile the code after downloading it, the first step is to extract the compressed build folder.

In Linux, the GPU and CPU codes are compressed using gzip and tar (\*.tar.gz). To extract, simply move to the desired folder and type in the command line:

```
$ tar -xzf <file name>.tar.gz
```



## COMPILING GOMC

GOMC generates four executable files for CPU code; GOMC\_CPU\_GEMC (Gibbs ensemble), GOMC\_CPU\_NVT (NVT ensemble), GOMC\_CPU\_NPT (isobaric-isothermal ensemble), and GOMC\_CPU\_GCMC (Grand canonical ensemble). In case of installing CUDA Toolkit, GOMC will generate additional four executable files for GPU code; GOMC\_GPU\_GEMC, GOMC\_GPU\_NVT, GOMC\_GPU\_NPT, and GOMC\_GPU\_GCMC.

Each ensemble has a respective unit test executable GOMC\_GEMC\_Test (Gibbs ensemble), GOMC\_NVT\_Test (NVT ensemble), GOMC\_NPT\_Test (isobaric-isothermal ensemble), and GOMC\_GCMC\_Test (Grand canonical ensemble). In case of installing CUDA Toolkit, GOMC will generate additional four unit test executables for GPU code; GOMC\_GPU\_GEMC\_Test, GOMC\_GPU\_NVT\_Test, GOMC\_GPU\_NPT\_Test, and GOMC\_GPU\_GCMC\_Test.

This section guides users to compile GOMC in Linux or Windows.

### 7.1 Linux

First, navigate your command line to the GOMC base directory. To compile GOMC on Linux, give permission to “metamake.sh” by running the following command:

```
$ chmod u+x metamake.sh
```

Metamake is the build script which creates a “bin” directory, configures and runs cmake file, and compiles the code as well. All executable files will be generated in the “bin” directory. By default, GOMC compiles all ensembles with the Intel compiler (icc), if available. Changes to this configuration can be made with options and arguments.

`./metamake.sh [OPTIONS] [ARGUMENTS]`

#### OPTIONS

<b>-d</b>	Compile in debug mode.
<b>-g</b>	Compile with gcc.
<b>-m</b>	Compile with MPI enabled.
<b>-p</b>	Compile with NVTX profiling for CUDA
<b>-t</b>	Compile Google tests.

#### ARGUMENTS

NVT NPT GCMC GEMC

If CUDA Toolkit found:

GPU\_NVT GPU\_NPT GPU\_GCMC GPU\_GEMC

If testing enabled:

```
GOMC_NVT_Test GOMC_NPT_Test GOMC_GCMC_Test GOMC_GEMC_Test
GOMC_GPU_NVT_Test      GOMC_GPU_NPT_Test      GOMC_GPU_GCMC_Test
GOMC_GPU_GEMC_Test
```

## 7.2 Windows

To compile GOMC on in Windows, follow these steps:

1. Open the Windows-compatible CMake GUI.
2. Set the Source Folder to the GOMC root folder.
3. Set the build Folder to your Build Folder.
4. Click configure, select your compiler/environment
5. Wait for CMake to finish the configuration.
6. Click configure again and click generate.
7. Open the CMake-generated project/solution etc. to the desired IDE (e.g Visual Studio).
8. Using the solution in the IDE of choice build GOMC per the IDE's standard release compilation/exe- cutable generation methods.

---

**Note:** You can also use CMake from the Windows command line if its directory is added to the PATH environment variable.

---

## 7.3 Configuring CMake

GOMC uses CMAKE to generate multi-platform intermediate files to compile the project. In this section, you can find all the information needed to configure CMake. We recommend using a different directory for the CMake output than the home directory of the project as CMake tend to generate lots of files.

We recommend configuring CMake through metamake.sh OPTIONS, but CMake has a ridiculously expansive set of options which are not all configurable through metamake. This document will only reproduce the most obviously relevant ones. When possible, options should be passed into CMake via command line options rather than the CMake-Cached.txt file:

**CMAKE\_BUILD\_TYPE** To get the best performance you should build the project in release mode. In CMake GUI you can set the value of “CMAKE\_BUILD\_TYPE” to “Release” and in CMake command line you can add the following to the CMake:

```
-DCMAKE_BUILD_TYPE=Release
```

To compile the GOMC in debug mode, in CMake GUI, change the value of “CMAKE\_BUILD\_TYPE” to “Debug” and in CMake command line you can add the following to the CMake:

```
-DCMAKE_BUILD_TYPE=Debug
```

Other options are “<None | ReleaseWithDebInfo | MinSizeRel>”.

**CMAKE\_CXX\_COMPILER** This option will set the compiler. It is recommended to use the Intel Compiler and linking tools, if possible (icc/icpc/etc.). They significantly outperform the default GNU and Visual Studio compiler tools and are available for free for academic use with registration.

**CMAKE\_CXX\_FLAGS\_RELEASE:STRING** To run the parallel version of CPU code, it needs to be compiled with openmp library. Open the file “CMakeCache.txt”, while still in the “bin” folder, and change the value from “-O3 -DNDEBUG” to “-O3 -qopenmp -DNDEBUG”. Recompile the GOMC by typing the command:

```
$ make
```

**ENSEMBLE\_NVT** You can turn the compilation of CPU version of NVT ensemble on or off using this option. - DENSEMBLE\_NVT=<On | Off>

**ENSEMBLE\_NPT** You can turn the compilation of CPU version of NPT ensemble on or off using this option. - DENSEMBLE\_NPT=<On | Off>

**ENSEMBLE\_GCMC** You can turn the compilation of CPU version of GCMC ensemble on or off using this option. -DENSEMBLE\_GCMC=<On | Off>

**ENSEMBLE\_GEMC** You can turn the compilation of CPU version of GEMC ensemble on or off using this option. -DENSEMBLE\_GEMC=<On | Off>

**ENSEMBLE\_GPU\_NVT** You can turn the compilation of GPU version of NVT ensemble on or off using this option. -DENSEMBLE\_NVT=<On | Off>

**ENSEMBLE\_GPU\_NPT** You can turn the compilation of GPU version of NPT ensemble on or off using this option. -DENSEMBLE\_NPT=<On | Off>

**ENSEMBLE\_GPU\_GCMC** You can turn the compilation of GPU version of GCMC ensemble on or off using this option. -DENSEMBLE\_GCMC=<On | Off>

**ENSEMBLE\_GPU\_GEMC** You can turn the compilation of GPU version of GEMC ensemble on or off using this option. -DENSEMBLE\_GEMC=<On | Off>



## GPU-ACCELERATED GOMC

**All moves use the following general GPU-Accelerated kernels:**

- All-molecule Intermolecular Lennard Jones and Coulombic Energy
- All-molecule Intermolecular Reciprocal Space Energy
- Image calculation for Ewald Summation
- Minimum Image Calculation

**GOMC currently supports several move-specific GPU-Accelerated kernels:**

- **Non-Equilibrium Molecule Transfer**
  - Single-molecule Reciprocal Space
  - Single-molecule Energy Change
- **Multi-Particle Moves:**
  - All-molecule Intermolecular Force (Lennard Jones and Coulombic)
  - All-molecule Intermolecular Reciprocal Space Force
  - **Force-biased MultiParticle**
    - \* All-molecule Force-biased Translation/Rotation
  - **Brownian-Motion MultiParticle**
    - \* All-molecule Brownian-motion Translation/Rotation





## INPUT FILE FORMATS

In order to run simulation in GOMC, the following files need to be provided:

- GOMC executable
- PDB file(s)
- PSF file(s)
- Parameter file
- Input file “NAME.conf” (proprietary control file)

In order to restart a simulation in GOMC from exactly where it left off, the following files also need to be provided:

- XSC file(s)
- COOR file(s)
- CHK file

In order to run a hybrid MCMD simulation, the following files also need to be provided:

- XSC file(s)
- COOR file(s)
- VEL files(s)
- CHK file

### 9.1 PDB File

GOMC requires only one PDB file for NVT and NPT ensembles. However, GOMC requires two PDB files for GEMC and GCMC ensembles.

#### 9.1.1 What is PDB file

The term PDB can refer to the Protein Data Bank (<http://www.rcsb.org/pdb/>), to a data file provided there, or to any file following the PDB format. Files in the PDB include various information such as the name of the compound, the ATOM and HETATM records containing the coordinates of the molecules, and etc. PDB widely used by NAMD, GROMACS, CHARMM, ACEMD, and Amber. GOMC ignore everything in a PDB file except for the REMARK, CRYST1, ATOM, and END records. An overview of the PDB standard can be found here:

<http://www.wwpdb.org/documentation/file-format-content/format33/sect2.html#HEADER>

<http://www.wwpdb.org/documentation/file-format-content/format33/sect8.html#CRYST1>

<http://www.wwpdb.org/documentation/file-format-content/format33/sect9.html#ATOM>

PDB contains four major parts; REMARK, CRYST1, ATOM, and END. Here is the definition of each field and how GOMC is using them to get the information it requires.

- **REMARK:** This header records present experimental details, annotations, comments, and information not included in other records (for more information, [click here](#)).

However, GOMC uses this header to print simulation informations.

- **Max Displacement** (Å)
- **Max Rotation** (Degree)
- **Max volume exchange** (<sup>3</sup>)
- **Monte Carlo Steps** (MC)
- **CRYST1:** This header records the unit [cell dimension parameters](#).
  - **Lattice constant:** a,b,c (Å)
  - **Lattice angles:**  $\alpha, \beta, \gamma$  (Degree)
- **ATOM:** The [ATOM](#) records present the atomic coordinates for standard amino acids and nucleotides. They also present the occupancy and temperature factor for each atom.
  - **ATOM:** Record name
  - **serial:** Atom serial number.
  - **name:** Atom name.
  - **resName:** Residue name.
  - **chainID:** Chain identifier.
  - **resSeq:** Residue sequence number.
  - **x:** Coordinates for X (Å).
  - **y:** Coordinates for Y (Å).
  - **z:** Coordinates for Z (Å).
  - **occupancy:** GOMC uses to define which atoms belong to which box.
  - **beta:** Beta or Temperature factor. GOMC uses this value to define the mobility of the atoms. element: Element symbol.
- **END:** A frame in the PDB file is terminated with the keyword.

Here are the PDB output of GOMC for the first molecule of isobutane:

REMARK	GOMC	122.790	3.14159	3439.817	1000000				
CRYST1	35.245	35.245	35.245	90.00	90.00	90.00			
ATOM	1	C1	ISB	1	0.911	-0.313	0.000	0.00	0.00 C
ATOM	2	C1	ISB	1	1.424	-1.765	0.000	0.00	0.00 C
ATOM	3	C1	ISB	1	-0.629	-0.313	0.000	0.00	0.00 C
ATOM	4	C1	ISB	1	1.424	0.413	-1.257	0.00	0.00 C
END									

The fields seen here in order from left to right are the record type, atom ID, atom name, residue name, residue ID, x, y, and z coordinates, occupancy, temperature factor (called beta), and segment name.

The atom name is “C1” and residue name is “ISB”. The PSF file (next section) contains a lookup table of atoms. These contain the atom name from the PDB and the name of the atom kind in the parameter file it corresponds to. As multiple

different atom names will all correspond to the same parameter, these can be viewed “atom aliases” of sorts. The chain letter (in this case ‘A’) is sometimes used when packing a number of PDBs into a single PDB file.

### Important:

- VMD requires a constant number of ATOMS in a multi-frame PDB (multiple records terminated by “END” in a single file). To compensate for this, all atoms from all boxes in the system are written to the output PDBs of this code.
- For atoms not currently in a box, the coordinates are set to  $\langle 0.00, 0.00, 0.00 \rangle$ . The occupancy is commonly just set to “1.00” and is left unused by many codes. We recycle this legacy parameter by using it to denote, in our output PDBs, the box a molecule is in (box 0 occupancy=0.00 ; box 1 occupancy=1.00)
- The beta value in GOMC code is used to define the mobility of the molecule.
  - Beta = 0.00: molecule can move and transfer within and between boxes.
  - Beta = 1.00: molecule is fixed in its position.
  - Beta = 2.00: molecule can move within the box but cannot be transferred between boxes.

## 9.1.2 Generating PDB file

With that overview of the format in mind, the following steps describe how a PDB file is typically built.

1. A single molecule PDB is obtained. In this example, the GaussView was used to draw the molecule, which was then edited by hand to adhere to the PDB spec properly. There are many open-source software that can build a molecule for you, such as [Avagadro](#), [molefactory](#) in VMD and more. The end result is a PDB for a single molecule:

```
REMARK    1 File created by GaussView 5.0.8
ATOM      1  C1  ISB  1    0.911  -0.313    0.000  C
ATOM      2  C1  ISB  1    1.424  -1.765    0.000  C
ATOM      3  C1  ISB  1   -0.629  -0.313    0.000  C
ATOM      4  C1  ISB  1    1.424    0.413   -1.257  C
END
```

2. Next, packings are calculated to place the simulation in a region of vapor-liquid coexistence. There are a couple of ways to do this in Gibbs ensemble:
  - Pack both boxes to a single middle density, which is an average of the liquid and vapor densities.
  - Same as previous method, but add a modest amount to axis of one box (e.g. 10-30 Å). This technique can be handy in the constant pressure Gibbs ensemble.
  - Pack one box to the predicted liquid density and the other to the vapor density.

A good reference for getting the information needed to estimate packing is the NIST Web Book database of pure compounds:

<http://webbook.nist.gov/chemistry/>

3. After packing is determined, a basic pack can be performed with a Packmol script. Here is the example of packing 1000 isobutane in 70 Å cubic box:

```
tolerance  3.0
filetype   pdb
output     STEP2_ISB_packed_BOX 0.pdb
```

(continues on next page)

(continued from previous page)

```

structure  isobutane.pdb
number     1000
inside cube 0.1  0.1  0.1  70.20
end        structure

```

Copy the above text into “pack\_isobutane.inp” file, save it and run the script by typing the following line into the terminal:

```
$ ./packmol < pack_isobutane.inp
```

## 9.2 PSF File

GOMC requires only one PSF file for NVT and NPT ensembles. However, GOMC requires two PSF files for GEMC and GCMC ensembles.

### 9.2.1 What is PSF file

Protein structure file (PSF), contains all of the molecule-specific information needed to apply a particular force field to a molecular system. The CHARMM force field is divided into a topology file, which is needed to generate the PSF file, and a parameter file, which supplies specific numerical values for the generic CHARMM potential function. The topology file defines the atom types used in the force field; the atom names, types, bonds, and partial charges of each residue type; and any patches necessary to link or otherwise mutate these basic residues. The parameter file provides a mapping between bonded and nonbonded interactions involving the various combinations of atom types found in the topology file and specific spring constants and similar parameters for all of the bond, angle, dihedral, improper, and van der Waals terms in the CHARMM potential function. PSF file widely used by by NAMD, CHARMM, and X-PLOR.

The PSF file contains six main sections: **remarks**, **atoms**, **bonds**, **angles**, **dihedrals**, and **impropers** (dihedral force terms used to maintain planarity). Each section starts with a specific header described below:

- **NTITLE**: remarks on the file. The following is taken from a PSF file for isobutane:

```

PSF
      3  !NTITLE
REMARKS  original generated structure x-plor psf file
REMARKS  topology ./Top_Branched_Alkanes.inp
REMARKS  segment ISB { first NONE; last NONE; auto angles dihedrals }

```

- **NATOM**: Defines the atom names, types, and partial charges of each residue type.

```

atom      ID
segment   name
residue    ID
residue    name
atom       name
atom       type
atom       charge
atom       mass

```

The following is taken from a PSF file for isobutane:

```

4000 !NATOM
1   ISB  1   ISB   C1   CH1   0.000000  13.0190  0
2   ISB  1   ISB   C2   CH3   0.000000  15.0350  0
3   ISB  1   ISB   C3   CH3   0.000000  15.0350  0
4   ISB  1   ISB   C4   CH3   0.000000  15.0350  0
5   ISB  2   ISB   C1   CH1   0.000000  13.0190  0
6   ISB  2   ISB   C2   CH3   0.000000  15.0350  0
7   ISB  2   ISB   C3   CH3   0.000000  15.0350  0
8   ISB  2   ISB   C4   CH3   0.000000  15.0350  0

```

The fields in the atom section, from left to right are atom ID, segment name, residue ID, residue name, atom name, atom type, charge, mass, and an unused 0.

- **NBOND:** The covalent bond section lists four pairs of atoms per line. The following is taken from a PSF file for isobutane:

```

3000 !BOND:      bonds
1   2   1   3   1   4   5   6
5   7   5   8

```

- **NTHETA:** The angle section lists three triples of atoms per line. The following is taken from a PSF file for isobutane:

```

3000 !NTHETA:    angles
2   1   4   2   1   3   3   1   4
6   5   8   6   5   7   7   5   8

```

- **NPHI:** The dihedral sections list two quadruples of atoms per line.
- **NIMPHI:** The improper sections list two quadruples of atoms per line. GOMC currently does not support improper. For the molecules without dihedral or improper, PDF file look like the following:

```

0   !NPHI:  dihedrals
0   !NIMPHI: impropers

```

- (other sections such as cross terms)

---

### Important:

- The PSF file format is a highly redundant file format. It repeats identical topology of thousands of molecules of a common kind in some cases. GOMC follows the same approach as NAMD, allowing this excess information externally and compiling it in the code.
  - Other sections (e.g. cross terms) contain unsupported or legacy parameters and are ignored.
  - Following the restriction of VMD, the order of the atoms in PSF file must match the order of the atoms in the PDB file.
  - Improper entries are read and stored, but are not currently used. Support will eventually be added for this.
-

### 9.2.2 Generating PSF file

The PSF file is typically generated using PSFGen. It is convenient to make a script, such as the example below, to do this:

```
package require psfgen
topology ./Top_branched_Alaknes.inp
segment ISB {
  pdb ./STEP2_ISB_packed_BOX 0.pdb
  first none
  last none
}

coordpdb ./STEP2_ISB_packed_BOX 0.pdb ISB

writepsf ./STEP3_START_ISB_sys_BOX_0.psf
writepdb ./STEP3_START_ISB_sys_BOX_0.pdb
```

Typically, one script is run per box to generate a finalized PDB/PSF for that box. The script requires one additional file, the NAMD-style topology file. While GOMC does not directly read or interact with this file, it's typically used to generate the PSF and, hence, is considered one of the integral file types. It will be briefly discussed in the following section.

## 9.3 Topology File

A CHARMM forcefield topology file contains all of the information needed to convert a list of residue names into a complete PSF structure file. The topology is a whitespace separated file format, which contains a list of atoms and their corresponding masses, and a list of residue information (charges, composition, and topology). Essentially, it is a non-redundant lookup table equivalent to the PSF file.

This is followed by a series of residues, which tell PSFGen what atoms are bonded to a given atom. Each residue is comprised of four key elements:

- A header beginning with the keyword RESI with the residue name and net charge
- A body with multiple ATOM entries (not to be confused with the PDB-style entries of the same name), which list the partial charge on the particle and what kind of atom each named atom in a specific molecule/residue is.
- A section of lines starting with the word BOND contains pairs of bonded atoms (typically 3 per line)
- A closing section with instructions for PSFGen.

Here's an example of topology file for isobutane:

```
* Custom top file -- branched alkanes *
11
!
MASS 1 CH3 15.035 C !
MASS 2 CH1 13.019 C !

AUTOGENERATE ANGLES DIHEDRALS

RESI ISB 0.00 ! isobutane - TraPPE
GROUP
ATOM C1 CH1 0.00 ! C3\
```

(continues on next page)

(continued from previous page)

```

ATOM  C2  CH3  0.00  !      C1-C2
ATOM  C3  CH3  0.00  !    C4/
ATOM  C4  CH3  0.00  !
BOND  C1  C2  C1  C3  C1  C4
PATCHING FIRS NONE LAST NONE

END

```

**Note:** The keyword END must be used to terminate this file and keywords related to the auto-generation process must be placed near the top of the file, after the MASS definitions.

**Tip:** More in-depth information can be found in the following links:

- [Topology Tutorial](#)
- [NAMD Tutorial: Examining the Topology File](#)
- [Developing Topology and Parameter Files](#)
- [NAMD Tutorial: Topology Files](#)

## 9.4 Parameter File

Currently, GOMC uses a single parameter file and the user has the two kinds of parameter file choices:

- CHARMM (Chemistry at Harvard Molecular Mechanics) compatible parameter file
- EXOTIC or Mie parameter file

If the parameter file type is not specified or if the chosen file is missing, an error will result.

Both force field file options are whitespace separated files with sections preceded by a tag. When a known tag (representing a molecular interaction in the model) is encountered, reading of that section of the force field begins. Comments (anything after a \* or !) and whitespace are ignored. Reading concludes when the end of the file is reached or another section tag is encountered.

### 9.4.1 CHARMM format parameter file

CHARMM contains a widely used model for describing energies in Monte Carlo and molecular dynamics simulations. It is intended to be compatible with other codes that use such a format, such as NAMD. See [here](#) for a general overview of the CHARMM force field.

Here's the basic CHARMM contributions that are supported in GOMC:

$$\begin{aligned}
 U_{\text{bond}} &= \sum_{\text{bonds}} K_b (b - b_0)^2 \\
 U_{\text{angle}} &= \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2 \\
 U_{\text{dihedral}} &= \sum_{\text{dihedrals}} K_\phi [1 + \cos(n\phi - \delta)] \\
 U_{\text{LJ}} &= \sum_{\text{nonbonded}} \epsilon_{ij} \left[ \left( \frac{R_{\text{min}_{ij}}}{r_{ij}} \right)^{12} - 2 \left( \frac{R_{\text{min}_{ij}}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{\epsilon r_{ij}}
 \end{aligned}$$

As seen above, the following are recognized, read and used:

- **BONDS** - Quadratic expression describing bond stretching based on bond length ( $b$ ) in Angstrom – Typically, it is ignored as bonds are rigid for Monte Carlo simulations.

---

**Note:** GOMC does not sample bond stretch. To ignore the relative bond energy, set the  $K_b$  to a large value i.e. “999999999999”.

---

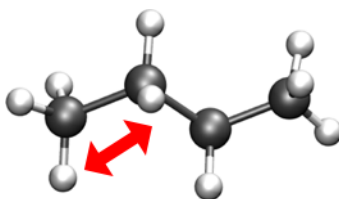


Fig. 1: Oscillations about the equilibrium bond length

- **ANGLES** - Describe the conformational behavior of an angle ( $\delta$ ) between three atoms, one of which is shared branch point to the other two.

---

**Note:** To fix any angle and ignore the related angle energy, set the  $K_\theta$  to a large value i.e. “999999999999”.

---

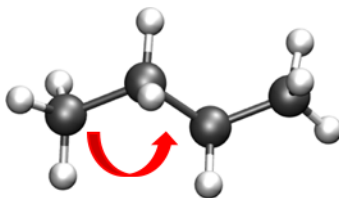


Fig. 2: Oscillations of 3 atoms about an equilibrium bond angle

- **DIHEDRALS** - Describes crankshaft-like rotation behavior about a central bond in a series of three consecutive bonds (rotation is given as  $\phi$ ).
- **NONBONDED** - This tag name only should be used if CHARMM force files are being used. This section describes 12-6 (Lennard-Jones) non-bonded interactions. Non-bonded parameters are assigned by specifying atom type name followed by polarizabilities (which will be ignored), minimum energy, and (minimum radius)/2. In order to modify 1-4 interaction, a second polarizability (again, will be ignored), minimum energy, and (minimum radius)/2 need to be defined; otherwise, the same parameter will be considered for 1-4 interaction.



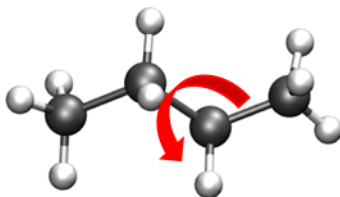


Fig. 3: Torsional rotation of 4 atoms about a central bond

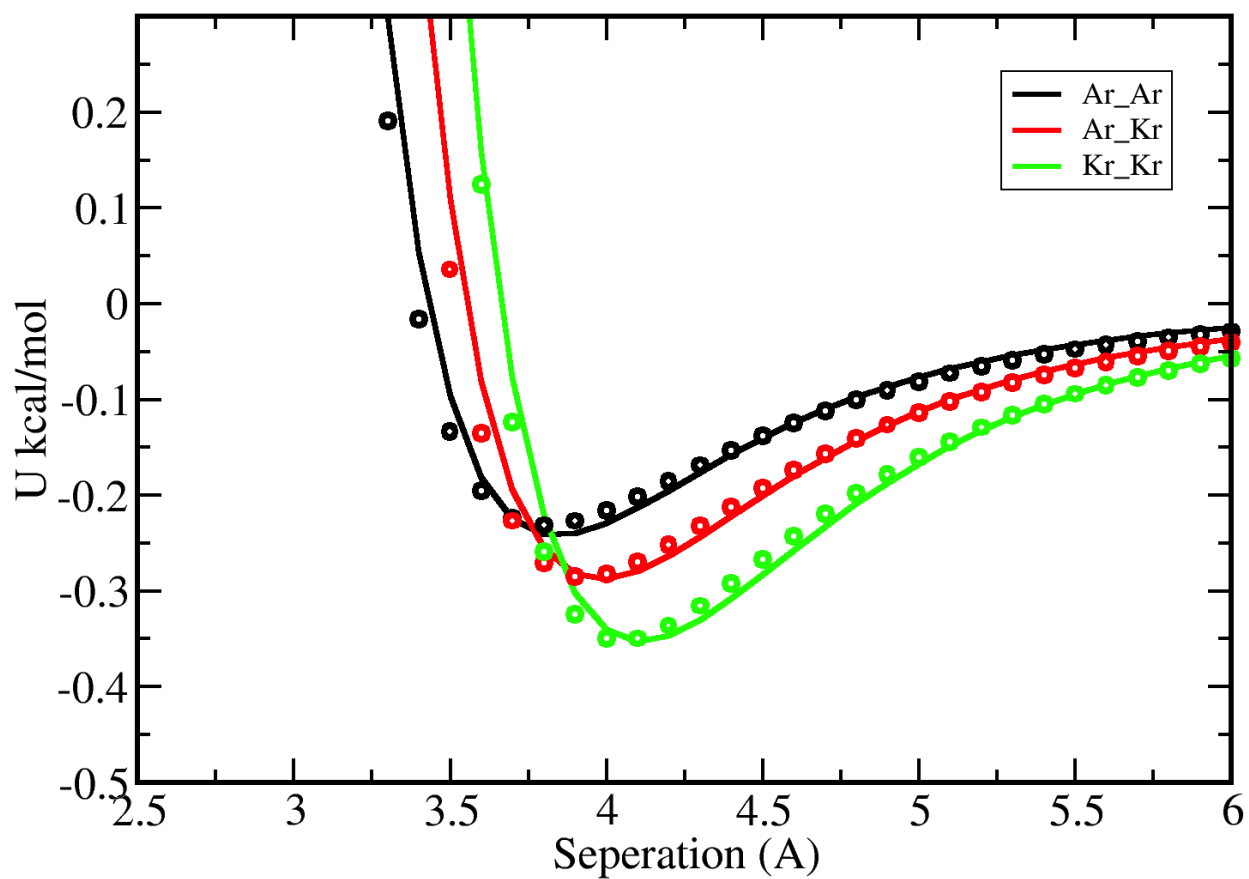


Fig. 4: Non-bonded energy terms (electrostatics and Lennard-Jones)

- **NBFIX** - This tag name only should be used if CHARMM force field is being used. This section allows interaction between two pairs of atoms to be modified, done by specifying two atom type names followed by minimum energy and minimum radius. In order to modify 1-4 interaction, a second minimum energy and minimum radius need to be defined; otherwise, the same parameter will be considered for 1-4 interaction.

---

**Note:** Please pay attention that in this section we define minimum radius, not (minimum radius)/2 as it is defined in the NONBONDED section.

---

Currently, supported sections of the CHARMM compliant file include BONDS, ANGLES, DIHEDRALS, NONBONDED, NBFIX. Other sections such as CMAP are not currently read or supported.

## 9.4.2 BONDS

(“bond stretching”) is one key section of the CHARMM-compliant file. Units for the  $K_b$  variable in this section are in kcal/mol; the  $b_0$  section (which represents the equilibrium bond length for that kind of pair) is measured in Angstroms.

$$U_{\text{bond}} = \sum_{\text{bonds}} K_b (b - b_0)^2$$

```

BONDS
!V(bond) = Kb(b - b0)**2
!
!Kb:  kcal/mole/A**2
!b0:  A
!
!Kb (kcal/mol) = Kb (K) * Boltz.  const.;
!
!atom type      Kb          b0      description
CH3   CH1   9999999999    1.540 !  TraPPE 2

```

---

**Note:** The  $K_b$  value may appear odd, but this is because a larger value corresponds to a more rigid bond. As Monte Carlo force fields (e.g. TraPPE) typically treat molecules as rigid constructs,  $K_b$  is set to a large value - 9999999999. Sampling bond stretch is not supported in GOMC.

---

## 9.4.3 ANGLES

(“bond bending”), where  $\theta$  is the measured bond angle and  $\theta_0$  is the equilibrium bond angle for that kind of pair, are commonly measured in degrees and  $K_\theta$  is the force constant measured in kcal/mol/K. These values, in literature, are often expressed in Kelvin (K).

To convert Kelvin to kcal/mol/K, multiply by the Boltzmann constant –  $K_\theta$ , 0.0019872041 kcal/mol. In order to fix the angle, it requires to set a large value for  $K_\theta$ . By assigning a large value like 9999999999, specified angle will be fixed and energy of that angle will be considered to be zero.

$$U_{\text{angle}} = \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2$$

Here is an example of what is necessary for isobutane:

```

ANGLES
!
!V(angle) = Ktheta(Theta - Theta0)**2
!
!V(Urey-Bradley) = Kub(S - S0)**2
!
!Ktheta: kcal/mole/rad**2
!Theta0: degrees
!S0: A
!
!Ktheta (kcal/mol) = Ktheta (K) * Boltz. const.
!
!atom types      Ktheta      Theta0
CH3  CH1  CH3      62.100125      112.00 ! TraPPE 2

```

Some CHARMM ANGLES section entries include Urey-Bradley potentials ( $K_{ub}$ ,  $b_{ub}$ ), in addition to the standard quadratic angle potential. The constants related to this potential function are currently read, but the logic has not been added to calculate this potential function. Support for this potential function will be added in later versions of the code.

## 9.4.4 DIHEDRALS

The final major bonded interactions section of the CHARMM compliant parameter file are the DIHEDRALS. Dihedral energies were represented by a cosine series where  $\phi$  is the dihedral angle,  $C_n$  are dihedral force constants,  $n$  is the multiplicity, and  $\delta_n$  is the phase shift. Often, there are 4 to 6 terms in a dihedral. Angles for the dihedrals' deltas are given in degrees.

$$U_{\text{dihedral}} = C_0 + \sum_{n=1} C_n [1 + \cos(n\phi_i - \delta_n)]$$

Since isobutane has no dihedral, here are the parameters pertaining to 2,3-dimethylbutane:

```

DIHEDRALS
!
!V(dihedral) = Kchi(1 + cos(n(chi) - delta))
!
!Kchi: kcal/mole
!n: multiplicity
!delta: degrees
!
!Kchi (kcal/mol) = Kchi (K) * Boltz. const.
!
!atom types      Kchi      n      delta      description
X  CH1  CH1  X    -0.498907  0      0.0      ! TraPPE 2
X  CH1  CH1  X     0.851974  1      0.0      ! TraPPE 2
X  CH1  CH1  X    -0.222269  2    180.0      ! TraPPE 2
X  CH1  CH1  X     0.876894  3      0.0      ! TraPPE 2

```

**Note:** The code allows the use of 'X' to indicate ambiguous positions on the ends. This is useful because this kind is often determined solely by the two middle atoms in the middle of the dihedral, according to literature.

**Note:** If a dihedral parameter was defined with multiplicity value of zero ( $n = 0$ ), GOMC will automatically assign

the phase shift value to 90 ( $\delta_n = 90$ ) to recover the above dihedral expression.

---

## 9.4.5 IMPROPERs

Energy parameters used to describe out-of-plane rocking are currently read, but unused. The section is often blank. If it becomes necessary, algorithms to calculate the improper energy will need to be added.

## 9.4.6 NONBONDED

The next section of the CHARMM style parameter file is the NONBONDED. The nonbonded energy in CHARMM is presented as 12-6 potential where,  $r_{ij}$ ,  $\epsilon_{ij}$ ,  $R_{minij}$  are the separation, minimum potential, and minimum potential distance, respectively. In order to use TraPPE this section of the CHARMM compliant file is critical.

$$U_{LJ} = \sum_{\text{nonbonded}} \epsilon_{ij} \left[ \left( \frac{R_{minij}}{r_{ij}} \right)^{12} - 2 \left( \frac{R_{minij}}{r_{ij}} \right)^6 \right]$$

Here's an example with our isobutane potential model:

```
NONBONDED
!
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!
!atom ignored epsilon      Rmin/2      ignored      eps,1-4      Rmin/2,1-4
CH3  0.0      -0.194745992  2.10461634058  0.0          0.0          0.0 ! TraPPE 1
CH1  0.0      -0.019872040  2.62656119304  0.0          0.0          0.0 ! TraPPE 2
End
```

---

**Note:** The  $R_{min}$  is the potential well-depth, where the attraction is maximum. However,  $\sigma$  is the particle diameter, where the interaction energy is zero. To convert  $\sigma$  to  $R_{min}$ , simply multiply  $\sigma$  by 0.56123102415.

---

---

**Important:** If no parameter was defined for 1-4 interaction e.g ( $\epsilon_{1-4}$ ,  $R_{min1-4}/2$ ), GOMC will use the  $\epsilon$ ,  $R_{min}/2$  for 1-4 interaction.

---

## 9.4.7 NBFIX

The last section of the CHARMM style parameter file is the NBFIX. In this section, individual pair interaction will be modified. First, pseudo non-bonded parameters have to be defined in NONBONDED and modified in NBFIX. Here is an example if it is required to modify interaction between CH3 and CH1 atoms:

```
NBFIX
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!
!atom atom  epsilon      Rmin      eps,1-4      Rmin,1-4
CH3  CH1  -0.294745992  1.10461634058 !
End
```

---

**Important:** If no parameter was defined for 1-4 interaction e.g ( $\epsilon_{1-4}$ ,  $R_{min_{1-4}}$ ), GOMC will use the  $\epsilon$ ,  $R_{min}$  for 1-4 interaction.

---

## 9.5 Exotic or Mie Parameter File

The Mie file is intended for use with nonstandard/specialty models of molecular interaction, which are not included in CHARMM standard.

### 9.5.1 Mie Potential

$$E_{ij} = C_{n_{ij}} \epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $\sigma_{ij}$  are, respectively, the separation, minimum potential, and collision diameter for the pair of interaction sites  $i$  and  $j$ . The constant  $C_n$  is a normalization factor such that the minimum of the potential remains at  $-\epsilon_{ij}$  for all  $n_{ij}$ . In the 12-6 potential,  $C_n$  reduces to the familiar value of 4.

$$C_{n_{ij}} = \left( \frac{n_{ij}}{n_{ij} - 6} \right) \left( \frac{n_{ij}}{6} \right)^{6/(n_{ij}-6)}$$

### 9.5.2 Buckingham Potential (Exp-6)

$$E_{ij} = \begin{cases} \frac{\alpha_{ij} \epsilon_{ij}}{\alpha_{ij} - 6} \left[ \frac{6}{\alpha_{ij}} \exp \left( \alpha_{ij} \left[ 1 - \frac{r_{ij}}{R_{min,ij}} \right] \right) - \left( \frac{R_{min,ij}}{r_{ij}} \right)^6 \right] & r_{ij} \geq R_{max,ij} \\ \infty & r_{ij} < R_{max,ij} \end{cases}$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $R_{min,ij}$  are, respectively, the separation, minimum potential, and minimum potential distance for the pair of interaction sites  $i$  and  $j$ . The constant  $\alpha_{ij}$  is an exponential-6 parameter. The cutoff distance  $R_{max,ij}$  is the smallest positive value for which  $\frac{dE_{ij}}{dr_{ij}} = 0$ .

---

**Note:** In order to use Mie or Exotice potential file format for Buckingham potential, instead of defining  $R_{min}$ , we define  $\sigma$  (collision diameter or the distance, where potential is zero) and GOMC will calculate the  $R_{min}$  and  $R_{max}$  using Buckingham potential equation.

---

Currently, two custom interaction are included:

- **NONBODED\_MIE** This section describes n-6 (Lennard-Jones) or Exp-6 (Buckingham) non-bonded interactions. The Lennard-Jones potential (12-6) is a subset of Mie potential. Non-bonded parameters are assigned by specifying the following fields in order:
  1. Atom type name
  2. Minimum energy ( $\epsilon$ )
  3. Atom diameter ( $\sigma$ )
  4. Repulsion exponent ( $n$ ) in Mie potential or  $\alpha$  in Buckingham potential.

The 1-4 interaction can be modified by specifying the following fields in order:

5. Minimum energy ( $\epsilon_{1-4}$ )

6. Atom diameter ( $\sigma_{1-4}$ )
7. Repulsion exponent ( $n_{1-4}$ ) in Mie potential or  $\alpha_{1-4}$  in Buckingham potential.

---

**Note:** If no parameter is provided for 1-4 interaction, same parameters (item 2, 3, 4) would be considered for 1-4 interaction.

---

- **NBFI\_X\_MIE** This section allows n-6 (Lennard-Jones) or Exp-6 (Buckingham) interaction between two pairs of atoms to be modified. Interaction between two pairs of atoms can be modified by specifying the following fields in order:

1. Atom type 1 name
2. Atom type 2 name
3. Minimum energy ( $\epsilon$ )
4. Atom diameter ( $\sigma$ )
5. Repulsion exponent ( $n$ ) in Mie potential or  $\alpha$  in Buckingham potential.

The 1-4 interaction between two pairs of atoms can be modified by specifying the following fields in order:

6. Minimum energy ( $\epsilon_{1-4}$ )
7. Atom diameter ( $\sigma_{1-4}$ )
8. Repulsion exponent ( $n_{1-4}$ ) in Mie potential or  $\alpha_{1-4}$  in Buckingham potential.

---

**Note:** If no parameter is provided for 1-4 interaction, same parameters (item 3, 4, 5) would be considered for 1-4 interaction.

---



---

**Note:** In Mie or Buckingham potential, the definition of atom diameter( $\sigma$ ) is same for both NONBONDED\_MIE and NBFI\_X\_MIE.

---



---

**Important:** If no parameter was defined for 1-4 interaction e.g ( $\epsilon_{1-4}$ ,  $\sigma_{1-4}$ ,  $n_{1-4}$ ), GOMC will use the  $\epsilon$ ,  $\sigma$ ,  $n$  for 1-4 interaction.

---

Otherwise, the Mie file reuses the same geometry section headings - BONDS / ANGLES / DIHEDRALS / etc. The only difference in these sections versus in the CHARMM format force field file is that the energies are in Kelvin ('K'), the unit most commonly found for parameters in Monte Carlo chemical simulation literature. This precludes the need to convert to kcal/mol, the energy unit used in CHARMM. The most frequently used section of the Mie files in the Mie potential section is NONBONDED\_MIE.

Here is the example of Mie or Exotic parameters file format that are used to simulate alkanes with Mie potential:

```
NONBONDED_MIE
!
!V(Mie) = const*eps*((sig/r)^n-(sig/r)^6)
!
!atom eps      sig      n      eps,1-4    sig,1-4    n,1-4
CH4  161.00    3.740    14      0.0        0.0        0.0 ! Potoff, et al. '09
CH3  121.25    3.783    16      0.0        0.0        0.0 ! Potoff, et al. '09
CH2   61.00    3.990    16      0.0        0.0        0.0 ! Potoff, et al. '09
```

(continues on next page)

(continued from previous page)

```

NBFIX_MIE
!V(Mie) = const*eps*((sig/r)^n-(sig/r)^6)
!
!atom atom  epsilon  sig      n      eps,1-4  sig,1-4  n,1-4
CH3  CH2    100.00   3.8      16      0.0      0.0      0.0 !
End

```

Here is the example of Mie or Exotic parameters file format that are used to simulate water with Buckingham potential:

```

NONBONDED_MIE
!
!V(exp-6) = ((eps-ij * alpha)/(alpha - 6)) * ((6 / alpha) * exp(alpha * [1 - (r /
↪rmin)])) - (rmin / r)^6))
!
!atom eps      sig      alpha      eps,1-4  sig,1-4  n,1-4
OT    159.78    3.195    12      0.0      0.0      0.0 ! Errington, et al. 1998
HT     0.0      0.0      0       0.0      0.0      0.0 ! Errington, et al. 1998

NBFIX_MIE
!V(exp-6) = ((eps-ij * alpha)/(alpha - 6)) * ((6 / alpha) * exp(alpha * [1 - (r /
↪rmin)])) - (rmin / r)^6))
!
!atom atom  epsilon  sig      alpha      eps,1-4  sig,1-4  n,1-4
HT  OT      0.00    0.0      0       0.0      0.0      0.0 !
End

```

**Note:** Although the units (Angstroms) are the same, the Mie file uses  $\sigma$ , not the  $R_{min}$  used by CHARMM. The energy in the exotic file are expressed in Kelvin (K), as this is the standard convention in the literature.

## 9.6 Control File (\*.conf)

The control file is GOMC's proprietary input file. It contains key settings. The settings generally fall under three categories:

- Input/Simulation Setup
- System Settings for During Run
- Output Settings

**Note:** The control file is designed to recognize logic values, such as “yes/true/on” or “no/false/off”. The keyword in control file is not case sensitive.

### 9.6.1 Input/Simulation Setup

In this section, input file names are listed. In addition, if you want to restart your simulation or use integer seed for running your simulation, you need to modify this section according to your purpose.

**Restart** Determines whether to restart the simulation from restart file (*\*\_restart.pdb*) or not.

- Value 1: Boolean - True if restart, false otherwise.

**ExpertMode** Determines whether to perform error checking of move selection to ensure correct ensemble is sampled. This allows the user to run a simulation with no volume moves in NPT, NPT-GEMC; no molecule transfers in GCMC, GEMC.

- Value 1: Boolean - True if enable expert mode; false otherwise.

**Checkpoint** Determines whether to restart the simulation from checkpoint file or not. Restarting the simulation with would result in an identical outcome, as if previous simulation was continued. This is required for hybrid Monte-Carlo Molecular Dynamics in open-ensembles (GCMC/GEMC) to concatenate trajectory files since the molecular transfers rearranges the order of the molecules. Checkpointing will ensure the molecules are loaded in the same order each cycle.

- Value 1: Boolean - True if restart with checkpoint file, false otherwise.
- Value 2: String - Sets the name of the checkpoint file.

```
Checkpoint    true          AR_KR_continued.chk
```

**PRNG** Dictates how to start the pseudo-random number generator (PRNG)

- Value 1: String
  - RANDOM: Randomizes Mersenne Twister PRNG with random bits based on the system time.

```
#####
# kind {RANDOM, INTSEED}
#####
PRNG    RANDOM
```

- INTSEED: This option “seeds” the Mersenne Twister PRNG with a standard integer. When the same integer is used, the generated PRNG stream should be the same every time, which is helpful in tracking down bugs.

**Random\_Seed** Defines the seed number. If “INTSEED” is chosen, seed number needs to be specified; otherwise, the program will terminate.

- Value 1: ULONG - If “INTSEED” option is selected for PRNG (See bellow example)

```
#####
# kind {RANDOM, INTSEED}
#####
PRNG          INTSEED
Random_Seed    50
```

**ParaTypeCHARMM** Sets force field type to CHARMM style.

- Value 1: Boolean - True if it is CHARMM forcefield, false otherwise.

```
#####
# FORCE FIELD TYPE
```

(continues on next page)



(continued from previous page)

```
#####
ParaTypeCHARMM    true
```

**ParaTypeEXOTIC or ParaTypeMie** Sets force field type to Mie style.

- Value 1: Boolean - True if it is Mie forcefield, false otherwise.

```
#####
# FORCE FIELD TYPE
#####
ParaTypeEXOTIC    true
```

**ParaTypeMARTINI** Sets force field type to MARTINI style.

- Value 1: Boolean - True if it is MARTINI forcefield, false otherwise.

```
#####
# FORCE FIELD TYPE
#####
ParaTypeMARTINI    true
```

**Parameters** Provides the name and location of the parameter file to use for the simulation.

- Value 1: String - Sets the name of the parameter file.

```
#####
# FORCE FIELD TYPE
#####
ParaTypeCHARMM    yes
Parameters         ../../common/Par_TraPPE_Alkanes.inp
```

**Coordinates** Defines the PDB file names (coordinates) and location for each box in the system.

- Value 1: Integer - Sets box number (starts from '0').
- Value 2: String - Sets the name of PDB file.

---

**Note:** NVT and NPT ensembles requires only one PDB file and GEMC/GCMC requires two PDB files. If the number of PDB files is not compatible with the simulation type, the program will terminate.

---

Example of NVT or NPT ensemble:

```
#####
# INPUT PDB FILES - NVT or NPT ensemble
#####
Coordinates    0    STEP3_START_ISB_sys.pdb
```

Example of Gibbs or GC ensemble:

```
#####
# INPUT PDB FILES - Gibbs or GCMC ensemble
#####
Coordinates    0    STEP3_START_ISB_sys_BOX_0.pdb
Coordinates    1    STEP3_START_ISB_sys_BOX_1.pdb
```

---

**Note:** In case of `Restart` true, the restart PDB output file from GOMC (`OutputName_BOX_N_restart.pdb`) can be used for each box.

---

Example of Gibbs ensemble when `Restart` mode is active:

```
#####  
# INPUT PDB FILES  
#####  
Coordinates 0 ISB_T_270_k_BOX_0_restart.pdb  
Coordinates 1 ISB_T_270_k_BOX_1_restart.pdb
```

**Structures** Defines the PSF filenames (structures) for each box in the system.

- Value 1: Integer - Sets box number (start from '0')
- Value 2: String - Sets the name of PSF file.

---

**Note:** NVT and NPT ensembles requires only one PSF file and GEMC/GCMC requires two PSF files. If the number of PSF files is not compatible with the simulation type, the program will terminate.

---

Example of NVT or NPT ensemble:

```
#####  
# INPUT PSF FILES  
#####  
Structure 0 STEP3_START_ISB_sys.psf
```

Example of Gibbs or GC ensemble:

```
#####  
# INPUT PSF FILES  
#####  
Structure 0 STEP3_START_ISB_sys_BOX_0.psf  
Structure 1 STEP3_START_ISB_sys_BOX_1.psf
```

---

**Note:** In case of `Restart` true, the PSF output file from GOMC (`OutputName_BOX_N_restart.psf`) can be used for both boxes.

---

Example of Gibbs ensemble when `Restart` mode is active:

```
#####  
# INPUT PSF FILES  
#####  
Structure 0 ISB_T_270_k_BOX_0_restart.psf  
Structure 1 ISB_T_270_k_BOX_1_restart.psf
```

**binCoordinates** Defines the DCD file names (coordinates) and location for each box in the system.

- Value 1: Integer - Sets box number (starts from '0').
- Value 2: String - Sets the name of PDB file.

---

**Note:** NVT and NPT ensembles requires only one DCD file and GEMC/GCMC requires only one PDB files, although loading two is supported. This is different from PDB files, for which two are required in GEMC/GCMC. This allows the user to only load binary coordinates for one box.

---

Example of NVT or NPT ensemble:

```
#####
# INPUT PDB FILES - NVT or NPT ensemble
#####
binCoordinates 0 STEP3_START_ISB_sys.coor
```

Example of Gibbs or GC ensemble:

```
#####
# INPUT PDB FILES - Gibbs or GCMC ensemble
#####
binCoordinates 0 STEP3_START_ISB_sys_BOX_0.coor
binCoordinates 1 STEP3_START_ISB_sys_BOX_1.coor
```

---

**Note:** In case of Restart, the restart DCD output file from GOMC (OutputName\_BOX\_N\_restart.coor) can be used for each box.

---

Example of Gibbs ensemble when Restart mode is active:

```
#####
# INPUT PDB FILES
#####
binCoordinates 0 ISB_T_270_k_BOX_0_restart.coor
binCoordinates 1 ISB_T_270_k_BOX_1_restart.coor
```

**binVelocities** Defines the VEL file names (velocities) and location for each box in the system.

- Value 1: Integer - Sets box number (starts from '0').
- Value 2: String - Sets the name of VEL file.

---

**Note:** Originate from a Molecular Dynamics software such as NAMD. GOMC will only output a velocity restart file if it is provided one using this keyword.

---



---

**Note:** In hybrid Monte-Carlo Molecular Dynamics, the velocities of the atoms should be preserved across cycles to increase accuracy. These files are not used internally by GOMC, only maintained. If a molecular transfer occurs, a new velocity is generated by Langevin dynamics.

---

Example of NVT or NPT ensemble:

```
#####
# INPUT PDB FILES - NVT or NPT ensemble
#####
binVelocities 0 STEP3_START_ISB_sys.vel
```

Example of Gibbs or GC ensemble:

```
#####  
# INPUT PDB FILES - Gibbs or GCMC ensemble  
#####  
binVelocities  0  STEP3_START_ISB_sys_BOX_0.vel  
binVelocities  1  STEP3_START_ISB_sys_BOX_1.vel
```

---

**Note:** In case of Restart, the restart VEL output file from GOMC (OutputName\_BOX\_N\_restart.vel) can be used for each box.

---

Example of Gibbs ensemble when Restart mode is active:

```
#####  
# INPUT PDB FILES  
#####  
binVelocities  0  ISB_T_270_k_BOX_0_restart.vel  
binVelocities  1  ISB_T_270_k_BOX_1_restart.vel
```

**extendedSystem** Defines the XSC file names (box dimensions and origin) and location for each box in the system.

- Value 1: Integer - Sets box number (starts from '0').
- Value 2: String - Sets the name of XSC file.

---

**Note:** Previously, this information was stored in plain-text format at the top of restart PDB files. This will be deprecated in favor of binary XSC.

---

Example of NVT or NPT ensemble:

```
#####  
# INPUT PDB FILES - NVT or NPT ensemble  
#####  
extendedSystem  0  STEP3_START_ISB_sys.xsc
```

Example of Gibbs or GC ensemble:

```
#####  
# INPUT PDB FILES - Gibbs or GCMC ensemble  
#####  
extendedSystem  0  STEP3_START_ISB_sys_BOX_0.xsc  
extendedSystem  1  STEP3_START_ISB_sys_BOX_1.xsc
```

---

**Note:** In case of Restart, the restart XSC output file from GOMC (OutputName\_BOX\_N\_restart.xsc) can be used for each box.

---

Example of Gibbs ensemble when Restart mode is active:

```
#####  
# INPUT PDB FILES  
#####
```

(continues on next page)

(continued from previous page)

extendedSystem	0	ISB_T_270_k_BOX_0_restart.xsc
extendedSystem	1	ISB_T_270_k_BOX_1_restart.xsc

**MultiSimFolderName** The name of the folder to be created which contains output from the multisim.

- Value 1: String - Name of the folder to contain output

MultiSimFolderName	outputFolderName
--------------------	------------------

## 9.7 XSC (eXtended System Configuration file) File

GOMC allows the box dimensions to be defined in one of three ways:

- In the control file
- In the header of restart PDB file
- In a binary XSC file

The XSC file contains the first step of the simulation, cell vectors, and cell origin. Currently, GOMC only uses the cell vectors.

## 9.8 COOR (binary coordinates) File

GOMC allows the box coordinates to be overwritten by a binary coordinates file. The COOR file should have the same number of atoms in it as the PDB file which it is overwriting. The actual coordinates can vary dramatically, which allows the user to sample the coordinates with other engines (MCMD), or transform it however one sees fit.

## 9.9 VEL (binary velocity) File

GOMC allows the velocities associated with each atom to be maintained and output for continuing MD simulations. In the event a molecule transfer occurs, all the atoms of the transferred molecule are given new velocities by Langevin dynamics. These VEL files must originate from NAMD, as GOMC will not produce them without first being provided them.

## 9.10 CHK (checkpoint) File

GOMC contains several variables which, if not accounted for, will produce different outputs even if the initial conditions are exactly the same. These variables are contained in the checkpoint file, and allow the user to pick up a GOMC simulation where it left off without altering the course of the simulation. Also, the checkpoint file is essential for MCMD as molecules are treated as distinguishable in molecular dynamics due to the fact that MD is a continuous trajectory through time. The checkpoint file contains the original atom order of the molecules, and coordinates and velocities are loaded into this order to ensure the trajectories are consistently arranged.

Checkpoint file contents:

- Last simulation step that saved into checkpoint file (Start step can be overridden).
- True number of simulation steps that have been run.

- Maximum amount of displacement ( $\text{\AA}$ ), rotation ( $\delta$ ), and volume ( $^3$ ) that used in Displacement, Rotation, MultiParticle, and Volume move.
- Number of Monte Carlo move trial and acceptance.
- Random number sequence.
- Molecule lookup object.
- Original pdb atoms object to reload new positions into.
- Original molecule setup object generated from parsing first PSF files.
- Accessory data for coordinating loading the restart coordinates into the original ordering.
- If built with MPI and parallel tempering was enabled: Random number sequence for parallel tempering.

### 9.10.1 System Settings for During Run Setup

This section contains all the variables not involved in the output of data during the simulation, or in the reading of input files at the start of the simulation. In other words, it contains settings related to the moves, the thermodynamic constants (based on choice of ensemble), and the length of the simulation. Note that some tags, or entries for tags, are only used in certain ensembles (e.g. Gibbs ensemble). These cases are denoted with colored text.

**GEMC** (*For Gibbs Ensemble runs only*) Defines the type of Gibbs Ensemble simulation you want to run. If neglected in Gibbs Ensemble, it simply defaults to const volume (NVT) Gibbs Ensemble.

- Value 1: String - Allows you to pick between isovolumetric (“NVT”) and isobaric (“NPT”) Gibbs ensemble simulations.

---

**Note:** The default value for GEMC is NVT.

---

```
#####  
# GEMC TYPE (DEFAULT IS NVT GEMC)  
#####  
GEMC      NVT
```

**Pressure** For NPT or NPT-GEMC simulation, imposed pressure (in bar) needs to be specified; otherwise, the program will terminate.

- Value 1: Double - Constant pressure in bar.

```
#####  
# GEMC TYPE (DEFAULT IS NVT GEMC)  
#####  
GEMC      NPT  
Pressure   5.76
```

**Temperature** Sets the temperature at which the system will run.

- Value 1: Double - Constant temperature of simulation in degrees Kelvin.

```
#####  
# SIMULATION CONDITION  
#####  
Temperature 270.00
```

(MPI-GOMC Only)

- Value 1: List of Doubles - A list of constant temperatures for simulations in degrees Kelvin.

```
#####
# SIMULATION CONDITION
#####
Temperature 270.00 280.00 290.00 300.00
```

---

**Note:** To use more than one temperature, GOMC must be compiled in MPI mode. Also, if GOMC is compiled in MPI mode, more than one temperature is required. To use only one temperature, use standard GOMC.

---

**Rcut** Sets a specific radius that non-bonded interaction energy and force will be considered and calculated using defined potential function.

- Value 1: Double - The distance to truncate the Lennard-Jones potential at.

**RcutLow** Sets a specific minimum possible in angstrom that reject any move that places any atom closer than specified distance.

- Value 1: Double - The minimum possible distance between any atoms.

**RcutCoulomb** Sets a specific radius for each box in the system that short range electrostatic energy will be calculated.

- Value 1: Integer - Sets box number (start from '0')
- Value 2: Double - The distance to truncate the short range electrostatic energy at.

---

**Note:** The default value for RcutCoulomb is the value of Rcut

---

#### Important:

- In Ewald Summation method, at constant Tolerance and box volume, increasing RcutCoulomb would result in decreasing reciprocal vector [Fincham 1993]. Decreasing the reciprocal vector decreases the computation time in long range electrostatic calculation.
  - Increasing the RcutCoulomb results in increasing the computation time in short range electrostatic calculation.
  - Parallelization of Ewald summation method is done on reciprocal vector loop, rather than molecule loop. So, in case of running on multiple CPU threads or GPU, it is better to use the lower value for RcutCoulomb, to maximize the parallelization efficiency.
  - There is an optimum value for RcutCoulomb, where result in maximum efficiency of the method. We encourage to run a short simulation with various RcutCoulomb to find the optimum value.
- 

**LRC** Defines whether or not long range corrections are used.

- Value 1: Boolean - True to consider long range correction.

---

**Note:** In case of using SHIFT or SWITCH potential functions, LRC will be ignored.

---

**Exclude** Defines which pairs of bonded atoms should be excluded from non-bonded interactions.

- Value 1: String - Allows you to choose between "1-2", "1-3", and "1-4".
  - 1-2: All interactions pairs of bonded atoms, except the ones that separated with one bond, will be considered and modified using 1-4 parameters defined in parameter file.

- 1-3: All interaction pairs of bonded atoms, except the ones that separated with one or two bonds, will be considered and modified using 1-4 parameters defined in parameter file.
- 1-4: All interaction pairs of bonded atoms, except the ones that separated with one, two or three bonds, will be considered using non-bonded parameters defined in parameter file.

---

**Note:** The default value for `Exclude` is “1-4”.

---



---

**Note:** In CHARMM force field, the 1-4 interaction needs to be considered. Choosing “`Exclude 1-3`” will modify 1-4 interaction based on 1-4 parameters in parameter file. If a kind force field is used, where 1-4 interaction needs to be ignored, such as TraPPE, either “`Exclude 1-4`” needs to be chosen or 1-4 parameter needs to be assigned to zero in the parameter file.

---

**Potential** Defines the potential function type to calculate non-bonded interaction energy and force between atoms.

- Value 1: String - Allows you to pick between “VDW”, “EXP6”, “SHIFT” and “SWITCH”.
  - VDW: Nonbonded interaction energy and force calculated based on n-6 (Lennard-Johns) equation. This function will be discussed further in the Intermolecular energy and Virial calculation section.

```
#####
# SIMULATION CONDITION
#####
Temperature 270.00
Potential    VDW
LRC          true
Rcut         10
Exclude      1-4
```

- EXP6: Nonbonded interaction energy and force calculated based on exp-6 (Buckingham potential) equation. This function will be discussed further in the Intermolecular energy and Virial calculation section.

```
#####
# SIMULATION CONDITION
#####
Temperature 270.00
Potential    EXP6
LRC          true
Rcut         10
Exclude      1-4
```

- SHIFT: This option forces the potential energy to be zero at `Rcut` distance. This function will be discussed further in the Intermolecular energy and Virial calculation section.

```
#####
# SIMULATION CONDITION
#####
Temperature 270.00
Potential    SHIFT
LRC          false
Rcut         10
Exclude      1-4
```

(continues on next page)



(continued from previous page)

```
RcutCoulomb 0 12.0
RcutCoulomb 1 20.0
```

- **SWITCH**: This option smoothly forces the potential energy to be zero at `Rcut` distance and starts modifying the potential at `Rswitch` distance. Depending on force field type, specific potential function will be applied. These functions will be discussed further in the Intermolecular energy and Virial calculation section.

**Rswitch** In the case of choosing “SWITCH” as potential function, a distance is set in which non-bonded interaction energy is truncated smoothly at `Rcut` distance.

- Value 1: Double - Define switch distance in angstrom. If the “SWITCH” function is chosen, `Rswitch` needs to be defined; otherwise, the program will be terminated.

```
#####
# SIMULATION CONDITION
#####
Temperature 270.00
Potential    SWITCH
LRC          false
Rcut         12
Rswitch      9
Exclude      1-4
```

**VDWGeometricSigma** Use geometric mean, as required by OPLS force field, to combining Lennard-Jones sigma parameters for different atom types.

- Value 1: Boolean - True, uses geometric mean to combine L-J sigmas

---

**Note:** The default setting of `VDWGeometricSigma` is false to use arithmetic mean when combining Lennard-Jones sigma parameters for different atom types.

---

**ElectroStatic** Considers coulomb interaction or not. This function will be discussed further in the Intermolecular energy and Virial calculation section.

- Value 1: Boolean - True if coulomb interaction needs to be considered and false if not.

---

**Note:** To simulate the polar molecule in MARTINI force field, `ElectroStatic` needs to be turn on. MARTINI force field uses short range coulomb interaction with constant Dielectric 15.0.

---

**Ewald** Considers standard Ewald summation method for electrostatic calculation. This function will be discussed further in the Intermolecular energy and Virial calculation section.

- Value 1: Double - True if Ewald summation calculation needs to be considered and false if not.

---

**Note:** By default, `ElectroStatic` will be set to true if Ewald summation method was used to calculate coulomb interaction.

---

**CachedFourier** Considers storing the reciprocal terms for Ewald summation calculation in order to improve the code performance. This option would increase the code performance with the cost of memory usage.

- Value 1: Boolean - True to store reciprocal terms of Ewald summation calculation and false if not.

---

**Note:** By default, `CachedFourier` will be set to true if not value was set.

---

**Warning:** Monte Carlo moves, such as MEMC-1, MEMC-2, MEMC-3, IntraMEMC-1, IntraMEMC-2, IntraMEMC-3 does not support `CachedFourier`.

**Tolerance** Specifies the accuracy of the Ewald summation calculation. Ewald separation parameter and number of reciprocal vectors for the Ewald summation are determined based on the accuracy parameter.

- Value 1: Double - Sets the accuracy in Ewald summation calculation.

---

**Note:**

- A reasonable value for the accuracy is 0.00001.
  - If “Ewald” was chosen and no value was set for `Tolerance`, the program will be terminated.
- 

**Dielectric** Defines dielectric constant for coulomb interaction in MARTINI force field.

- Value 1: Double - Sets dielectric value used in coulomb interaction.

---

**Note:**

- In MARTINI force field, `Dielectric` needs to be set to 15.0.
  - If MARTINI force field was chosen and `Dielectric` was not specified, a default value of 15.0 will be assigned.
- 

**PressureCalc** Considers to calculate the pressure or not. If it is set to true, the frequency of pressure calculation need to be set.

- Value 1: Boolean - True enabling pressure calculation during the simulation, false disabling pressure calculation.
- Value 2: Ulong - The frequency of calculating the pressure.

**1-4scaling** Defines constant factor to modify intra-molecule coulomb interaction.

- Value 1: Double - A fraction number between 0.0 and 1.0.

---

**Note:** CHARMM force field uses a value between 0.0 and 1.0. In MARTINI force field, it needs to be set to 1.0 because 1-4 interaction will not be modified in this force field.

---

```
#####  
# SIMULATION CONDITION  
#####  
ElectroStatic    true  
Ewald            true  
Tolerance        0.00001  
CachedFourier    false  
1-4scaling       0.0
```

**RunSteps** Sets the total number of steps to run (one move is performed for each step) (cycles = this value / number of molecules in the system)

- Value 1: Ulong - Total run steps

---

**Note:** RunSteps is a delta.

---



---

**Important:** Setting the RunSteps to zero, and activating Restart simulation, will recalculate the energy of stored simulation's snapshots.

---

**EqSteps** Sets the number of steps necessary to equilibrate the system; averaging will begin at this step.

- Value 1: Ulong - Equilibration steps

---

**Note:** EqSteps is not a delta. If restarting a simulation with a start step greater than EqSteps, no equilibration is performed.

---



---

**Note:** In GCMC simulation, the Histogram files will be outputted at EqSteps.

---

**AdjSteps** Sets the number of steps per adjustment of the parameter associated with each move (e.g. maximum translate distance, maximum rotation, maximum volume exchange, etc.)

- Value 1: Ulong - Number of steps per move adjustment

**InitStep** Sets the first step of the simulation.

- Value 1: Ulong - Number of first step of simulation.

---

**Note:** Hybrid Monte-Carlo Molecular Dynamics (py-MCMD) requires resetting start step to 0 for combination of NAMD and GOMC data.

---

```
#####
# STEPS
#####
RunSteps    25000000
EqSteps     5000000
AdjSteps    1000
InitStep     0
```

---

**ChemPot** For Grand Canonical (GC) ensemble runs only: Chemical potential at which simulation is run.

- Value 1: String - The residue name to apply this chemical potential.
- Value 2: Double - The chemical potential value in degrees Kelvin (should be negative).

---

**Note:**

- For binary systems, include multiple copies of the tag (one per residue kind).
  - If there is a molecule kind that cannot be transfer between boxes (in PDB file the beta value is set to 1.00 or 2.00), an arbitrary value (e.g. 0.00) can be assigned to the residue name.
-

```
#####  
# Mol. Name Chem. Pot. (K)  
#####  
ChemPot   ISB      -968
```

**Fugacity** For Grand Canonical (GC) ensemble runs only: Fugacity at which simulation is run.

- Value 1: String - The residue to apply this fugacity.
- Value 2: Double - The fugacity value in bar.

---

**Note:**

- For binary systems, include multiple copies of the tag (one per residue kind).
- If there is a molecule kind that cannot be transfer between boxes (in PDB file the beta value is set to 1.00 or 2.00) an arbitrary value e.g. 0.00 can be assigned to the residue name.

---

```
#####  
# Mol. Name Fugacity (bar)  
#####  
Fugacity ISB    10.0  
Fugacity Si      0.0  
Fugacity 0       0.0
```

**DisFreq** Fractional percentage at which displacement move will occur.

- Value 1: Double - % Displacement

**RotFreq** Fractional percentage at which rigid rotation move will occur.

- Value 1: Double - % Rotation

**IntraSwapFreq** Fractional percentage at which molecule will be removed from a box and inserted into the same box using coupled-decoupled configurational-bias algorithm.

- Value 1: Double - % Intra molecule swap

---

**Note:** The default value for IntraSwapFreq is 0.000

---

**IntraTargetedSwapFreq** Fractional percentage at which molecule will be removed from the box and inserted into a subvolume within the same box, or deleted from the subvolume and inserted into the same box using coupled-decoupled configurational-bias algorithm.

- Value 1: Double - % Intra molecule swap

---

**Note:** The default value for IntraTargetedSwapFreq is 0.000

---

**RegrowthFreq** Fractional percentage at which part of the molecule will be deleted and then regrown using coupled-decoupled configurational-bias algorithm.

- Value 1: Double - % Molecular growth

---

**Note:** The default value for RegrowthFreq is 0.000

---

**CrankShaftFreq** Fractional percentage at which crankshaft move will occur. In this move, two atoms that are forming angle or dihedral are selected randomly and form a shaft. Then any atoms or group that are within these two selected atoms, will rotate around the shaft to sample intramolecular degree of freedom.

- Value 1: Double - % Crankshaft

---

**Note:** The default value for CrankShaftFreq is 0.000

---

**MultiParticleFreq** Fractional percentage at which multi-particle move will occur. In this move, all molecules in the selected simulation box will be rigidly rotated or displaced simultaneously, along the calculated torque or force, respectively.

- Value 1: Double - % Multiparticle

---

**Note:** The default value for MultiParticleFreq is 0.000

---

**MultiParticleBrownianFreq** Fractional percentage at which multi-particle brownian move will occur. In this move, all molecules in the selected simulation box will be rigidly rotated or displaced simultaneously, along the calculated torque or force, respectively.

- Value 1: Double - % Multiparticle

---

**Note:** The default value for MultiParticleBrownianFreq is 0.000

---

**NeMTMCFreq** Fractional percentage at which non-equilibrium molecule transfer move will occur. In this move, a molecule is gradually transferred from the selected simulation box to the destination box, with the multi-particle move or multi-particle brownian move used to relax the system.

- Value 1: Double - % Non-equilibrium molecule transfer

---

**Note:** The default value for NeMTMCFreq is 0.000

---



---

**Note:** The number of RelaxingSteps per NeMTMC move must be defined.

---



---

**Note:** Either MultiParticleRelaxing or MultiParticleBrownianRelaxing must be enabled if NeMTMC move is to be used.

---



---

**Note:** ScalePower, ScaleAlpha, MinSigma, ScaleCoulomb parameters discussed in Free Energy section are used by NeMTMC moves.

---

**RelaxingSteps** Sets the total number of relaxing steps to run (one MP or BMP is performed for each step) to relax the system.

- Value 1: Ulong - Total relaxing steps per NeMTMC move

**MultiParticleRelaxing** Relax NeMTMC using force-biased Monte Carlo algorithm.

- Value 1: Boolean

---

**Note:** MultiParticleFreq must be non-zero if NeMTMC with MultiParticleRelaxing is to be used.

---

**MultiParticleBrownianRelaxing** Relax NeMTMC using brownian motion.

- Value 1: Boolean

---

**Note:** MultiParticleBrownianFreq must be non-zero if NeMTMC with MultiParticleBrownianRelaxing is to be used.

---

**SampleConfFreq** Intra-Swap/Regrowth Frequency in NeMTMC Relaxing Steps

- Value 1: Double

**LambdaVDWLimit** Lambda VDW limit for Intra-Swap move in NeMTMC Relaxing Steps

**IntraMEMC-1Freq** Fractional percentage at which specified number of small molecule kind will be exchanged with a specified large molecule kind in defined sub-volume within same simulation box.

- Value 1: Double - % Molecular exchange

---

**Note:**

- The default value for IntraMEMC-1Freq is 0.000
  - This move need additional information such as ExchangeVolumeDim, ExchangeRatio, ExchangeSmallKind, and ExchangeLargeKind, which will be explained later.
  - For more information about this move, please refere to [MEMC-GCMC](#) and [MEMC-GEMC](#) papers.
- 

**IntraMEMC-2Freq** Fractional percentage at which specified number of small molecule kind will be exchanged with a specified large molecule kind in defined sub-volume within same simulation box. Backbone of small and large molecule kind will be used to insert the large molecule more efficiently.

- Value 1: Double - % Molecular exchange

---

**Note:**

- The default value for IntraMEMC-2Freq is 0.000
  - This move need additional information such as ExchangeVolumeDim, ExchangeRatio, ExchangeSmallKind, ExchangeLargeKind, SmallKindBackBone, and LargeKindBackBone, which will be explained later.
  - For more information about this move, please refere to [MEMC-GCMC](#) and [MEMC-GEMC](#) papers.
- 

**IntraMEMC-3Freq** Fractional percentage at which specified number of small molecule kind will be exchanged with a specified large molecule kind in defined sub-volume within same simulation box. Specified atom of the large molecule kind will be used to insert the large molecule using coupled-decoupled configurational-bias.

- Value 1: Double - % Molecular exchange

---

**Note:**

- The default value for IntraMEMC-3Freq is 0.000

- This move need additional information such as `ExchangeVolumeDim`, `ExchangeRatio`, `ExchangeSmallKind`, `ExchangeLargeKind`, and `LargeKindBackBone`, which will be explained later.
- For more information about this move, please refer to [MEMC-GCMC](#) and [MEMC-GEMC](#) papers.

**MEMC-1Freq** For Gibbs and Grand Canonical (GC) ensemble runs only: Fractional percentage at which specified number of small molecule kind will be exchanged with a specified large molecule kind in defined sub-volume in dense simulation box.

- Value 1: Double - % Molecular exchange

**Note:**

- The default value for **MEMC-1Freq** is 0.000
- This move need additional information such as `ExchangeVolumeDim`, `ExchangeRatio`, `ExchangeSmallKind`, and `ExchangeLargeKind`, which will be explained later.
- For more information about this move, please refer to [MEMC-GCMC](#) and [MEMC-GEMC](#) papers.

**MEMC-2Freq** For Gibbs and Grand Canonical (GC) ensemble runs only: Fractional percentage at which specified number of small molecule kind will be exchanged with a specified large molecule kind in defined sub-volume in dense simulation box. Backbone of small and large molecule kind will be used to insert the large molecule more efficiently.

- Value 1: Double - % Molecular exchange

**Note:**

- The default value for **MEMC-2Freq** is 0.000
- This move need additional information such as `ExchangeVolumeDim`, `ExchangeRatio`, `ExchangeSmallKind`, `ExchangeLargeKind`, `SmallKindBackBone`, and `LargeKindBackBone`, which will be explained later.
- For more information about this move, please refer to [MEMC-GCMC](#) and [MEMC-GEMC](#) papers.

**MEMC-3Freq** For Gibbs and Grand Canonical (GC) ensemble runs only: Fractional percentage at which specified number of small molecule kind will be exchanged with a specified large molecule kind in defined sub-volume in dense simulation box. Specified atom of the large molecule kind will be used to insert the large molecule using coupled-decoupled configurational-bias.

- Value 1: Double - % Molecular exchange

**Note:**

- The default value for **MEMC-3Freq** is 0.000
- This move need additional information such as `ExchangeVolumeDim`, `ExchangeRatio`, `ExchangeSmallKind`, `ExchangeLargeKind`, and `LargeKindBackBone`, which will be explained later.
- For more information about this move, please refer to [MEMC-GCMC](#) and [MEMC-GEMC](#) papers.

**SwapFreq** For Gibbs and Grand Canonical (GC) ensemble runs only: Fractional percentage at which molecule swap move will occur using coupled-decoupled configurational-bias.

- Value 1: Double - % Molecule swaps

**TargetedSwapFreq** For Gibbs and Grand Canonical (GC) ensemble runs only: Fractional percentage at which targeted molecule swap move will occur using coupled-decoupled configurational-bias in the sub-volumes specified.

- Value 1: Double - % Molecule targeted swaps

**VolFreq** For isobaric-isothermal ensemble and Gibbs ensemble runs only: Fractional percentage at which molecule will be removed from one box and inserted into the other box using configurational bias algorithm.

- Value 1: Double - % Volume swaps

```
#####  
# MOVE FREQUENCY  
#####  
DisFreq      0.39  
RotFreq      0.10  
IntraSwapFreq 0.10  
RegrowthFreq 0.10  
CrankShaftFreq 0.10  
SwapFreq     0.20  
VolFreq      0.01
```

**Warning:** All move percentages should add up to 1.0; otherwise, the program will terminate.

**ExchangeVolumeDim** To use all variation of MEMC and IntraMEMC Monte Carlo moves, the exchange sub-volume must be defined. The exchange sub-volume is defined as an orthogonal box with x-, y-, and z-dimensions, where small molecule/molecules kind will be selected from to be exchanged with a large molecule kind.

- Value 1: Double - X dimension in
- Value 2: Double - Y dimension in
- Value 3: Double - Z dimension in

---

**Note:**

- Currently, the X and Y dimension cannot be set independently ( $X = Y = \max(X, Y)$ )
  - A heuristic for setting good values of the x-, y-, and z-dimensions is to use the geometric size of the large molecule plus 1-2 Å in each dimension.
  - In case of exchanging 1 small molecule kind with 1 large molecule kind in IntraMEMC-2, IntraMEMC-3, MEMC-2, MEMC-3 Monte Carlo moves, the sub-volume dimension has no effect on acceptance rate.
- 

**ExchangeSmallKind** To use all variation of MEMC and IntraMEMC Monte Carlo moves, the small molecule kind to be exchanged with a large molecule kind must be defined. Multiple small molecule kind can be specified.

- Value 1: String - Small molecule kind (resname) to be exchanged.

**ExchangeLargeKind** To use all variation of MEMC and IntraMEMC Monte Carlo moves, the large molecule kind to be exchanged with small molecule kind must be defined. Multiple large molecule kind can be specified.

- Value 1: String - Large molecule kind (resname) to be exchanged.

**ExchangeRatio** To use all variation of MEMC and IntraMEMC Monte Carlo moves, the exchange ratio must be defined. The exchange ratio defines how many small molecule will be exchanged with 1 large molecule. For each large-small molecule pairs, one exchange ratio must be defined.



- Value 1: Integer - Ratio of exchanging small molecule/molecules with 1 large molecule.

**LargeKindBackBone** To use MEMC-2, MEMC-3, IntraMEMC-2, and IntraMEMC-3 Monte Carlo moves, the large molecule backbone must be defined. The backbone of the molecule is defined as a vector that connects two atoms belong to the large molecule. The large molecule backbone will be used to align the sub-volume in MEMC-2 and IntraMEMC-2 moves, while in MEMC-3 and IntraMEMC-3 moves, it uses the atom name to start growing the large molecule using coupled-decoupled configurational-bias. For each large-small molecule pairs, two atom names must be defined.

- Value 1: String - Atom name 1 belong to the large molecule's backbone
- Value 2: String - Atom name 2 belong to the large molecule's backbone

---

**Important:**

- In MEMC-3 and IntraMEMC-3 Monte Carlo moves, both atom names must be same, otherwise program will be terminated.
  - If large molecule has only one atom (mono atomic molecules), same atom name must be used for Value 1 and Value 2 of the LargeKindBackBone.
- 

**SmallKindBackBone** To use MEMC-2, and IntraMEMC-2 Monte Carlo moves, the small molecule backbone must be defined. The backbone of the molecule is defined as a vector that connects two atoms belong to the small molecule and will be used to align the sub-volume. For each large-small molecule pairs, two atom names must be defined.

- Value 1: String - Atom name 1 belong to the small molecule's backbone
- Value 2: String - Atom name 2 belong to the small molecule's backbone

---

**Important:**

- If small molecule has only one atom (mono atomic molecules), same atom name must be used for Value 1 and Value 2 of the SmallKindBackBone.
- 

Here is the example of MEMC-2 Monte Carlo moves, where 7 large-small molecule pairs are defined with an exchange ratio of 1:1: (ethane, methane), (propane, ethane), (n-butane, propane), (n-pentane, nbutane), (n-hexane, n-pentane), (n-heptane, n-hexane), and (noctane, n-heptane).

```
#####
# MEMC PARAMETER
#####
ExchangeVolumeDim  1.0  1.0  1.0
ExchangeRatio      1    1    1    1    1    1    1
ExchangeLargeKind  C8P  C7P  C6P  C5P  C4P  C3P  C2P
ExchangeSmallKind  C7P  C6P  C5P  C4P  C3P  C2P  C1P
LargeKindBackBone  C1 C8 C1 C7 C1 C6 C1 C5 C1 C4 C1 C3 C1 C2
SmallKindBackBone  C1 C7 C1 C6 C1 C5 C1 C4 C1 C3 C1 C2 C1 C1
```

Here is the example of MEMC-3 Monte Carlo moves, where 7 large-small molecule pairs are defined with an exchange ratio of 1:1: (ethane, methane), (propane, ethane), (n-butane, propane), (n-pentane, nbutane), (n-hexane, n-pentane), (n-heptane, n-hexane), and (noctane, n-heptane).

```
#####
# MEMC PARAMETER
```

(continues on next page)

(continued from previous page)

```
#####
ExchangeVolumeDim  1.0  1.0  1.0
ExchangeRatio      1      1      1      1      1      1      1
ExchangeLargeKind  C8P   C7P   C6P   C5P   C4P   C3P   C2P
ExchangeSmallKind  C7P   C6P   C5P   C4P   C3P   C2P   C1P
LargeKindBackBone  C4 C4  C4 C4  C3 C3  C3 C3  C2 C2  C2 C2  C1 C1
SmallKindBackBone  C1 C7  C1 C6  C1 C5  C1 C4  C1 C3  C1 C2  C1 C1
```

Here is the example of MEMC-2 Monte Carlo moves, where 1 large-small molecule pair is defined with an exchange ratio of 1:2 (xenon, methane).

```
#####
# MEMC PARAMETER
#####
ExchangeVolumeDim  5.0  5.0  5.0
ExchangeRatio      2
ExchangeLargeKind  XE
ExchangeSmallKind  C1P
LargeKindBackBone  Xe Xe
SmallKindBackBone  C1 C1
```

**SubVolumeBox** Define which box the dynamic subvolume occupies. - Value 1: Integer - Sub-volume id. - Value 2: Integer - Sets box number (first box is box '0').

**SubVolumeCenter** Define the center of the dynamic subvolume. - Value 1: Integer - Sub-volume id. - Value 2: Double - x value of SubVolumeCenter. - Value 3: Double - y value of SubVolumeCenter. - Value 4: Double - z value of SubVolumeCenter.

**SubVolumePBC** Define which dimensions periodic box wrapping is applied in the subvolume. - Value 1: Integer - Sub-volume id. - Value 2: String - X, Y, Z, XY, XZ, YZ, XYZ (Axes should have no spaced between them)

**SubVolumeCenterList** Define the center of the subvolume by defining the atoms to use for the geometric mean calculation. - Value 1: Integer - Sub-volume id. - Value 2: Integer Range - Atom indices used to calculate geometric center of subvolume.

**SubVolumeDim** Define the dimensions of the subvolume. - Value 1: Integer - Sub-volume id. - Value 2: Double - x value of SubVolumeDim. - Value 3: Double - y value of SubVolumeDim. - Value 4: Double - z value of SubVolumeDim.

**SubVolumeResidueKind** Define which residue kinds can be inserted or deleted from the subvolume. - Value 1: Integer - Sub-volume id. - Value 2: String - Residue kind inserted/deleted from subvolume - Value 3: String - Residue kind inserted/deleted from subvolume - Value 4: String - Residue kind inserted/deleted from subvolume - Value N: String - Residue kind inserted/deleted from subvolume

**SubVolumeRigidSwap** Define whether molecules are held rigid or the geometry is sampled per the coupled-decoupled CBMC scheme. - Value 1: Integer - Sub-volume id. - Value 2: Boolean - If true the molecule is held rigid. If false, geometry is sampled when inserting in the subvolume.

**SubVolumeChemPot** Define the chemical potential of a residue kind in the subvolume. Only used in TargetedSwap, not IntraTargetedSwap. - Value 1: Integer - Sub-volume id. - Value 2: String - Residue kind - Value 3: Double - Chemical potential

**SubVolumeFugacity** Define the fugacity of a residue kind in the subvolume. Only used in TargetedSwap, not IntraTargetedSwap. - Value 1: Integer - Sub-volume id. - Value 2: String - Residue kind - Value 3: Double - Chemical potential

```
#####
# TARGETED SWAP (Dynamic subVolume)
#####
SubVolumeBox          1      0
SubVolumeCenterList    1      1-402
SubVolumeDim           1      35 35 5
SubVolumeResidueKind   1      TIP3
SubVolumeRigidSwap     1      false
```

**useConstantArea** For Isobaric-Isothermal ensemble and Gibbs ensemble runs only: Considers to change the volume of the simulation box by fixing the cross-sectional area (x-y plane).

- Value 1: Boolean - If true volume will change only in z axis, If false volume will change with constant axis ratio.

---

**Note:** By default, useConstantArea will be set to false if no value was set. It means, the volume of the box will change in a way to maintain the constant axis ratio.

---

**FixVolBox0** For adsorption simulation in NPT Gibbs ensemble runs only: Changing the volume of fluid phase (Box 1) to maintain the constant imposed pressure and temperature, while keeping the volume of adsorbed phase (Box 0) fix.

- Value 1: Boolean - If true volume of adsorbed phase will remain constant, If false volume of adsorbed phase will change.

**CellBasisVector** Defines the shape and size of the simulation periodic cell. CellBasisVector1, CellBasisVector2, CellBasisVector3 represent the cell basis vector  $a, b, c$ , respectively. This tag may occur multiple times. It occurs once for NVT and NPT, but twice for Gibbs ensemble or GC ensemble.

- Value 1: Integer - Sets box number (first box is box '0').
- Value 2: Double - x value of cell basis vector .
- Value 3: Double - y value of cell basis vector .
- Value 4: Double - z value of cell basis vector .

---

**Note:** If the number of defined boxes were not compatible to simulation type, the program will be terminated.

---

Example for NVT and NPT ensemble. In this example, each vector is perpendicular to the other two ( $\alpha = 90, \beta = 90, \gamma = 90$ ), as indicated by a single x, y, or z value being specified by each and making a rectangular 3-D box:

```
#####
# BOX DIMENSION #, X, Y, Z
#####
CellBasisVector1  0   40.00   00.00   00.00
CellBasisVector2  0   00.00   40.00   00.00
CellBasisVector3  0   00.00   00.00   80.00
```

Example for Gibbs ensemble and GC ensemble ensemble. In this example, In the first box, only vector  $a$  and  $c$  are perpendicular to each other ( $\alpha = 90, \beta = 90, \gamma = 120$ ), and making a non-orthogonal simulation cell with the cell length  $a = 39.91, b = 39.91, c = 76.98$ . In the second box, each vector is perpendicular to the other two ( $\alpha = 90, \beta = 90, \gamma = 90$ ), as indicated by a single x, y, or z value being specified by each and making a cubic box:

```
#####
# BOX DIMENSION #, X, Y, Z
#####
CellBasisVector1  0   36.91   00.00   00.00
CellBasisVector2  0  -18.45   31.96   00.00
CellBasisVector3  0   00.00   00.00   76.98

CellBasisVector1  1   60.00   00.00   00.00
CellBasisVector2  1   00.00   60.00   00.00
CellBasisVector3  1   00.00   00.00   60.00
```

**Warning:** If Restart was activated, box dimension does not need to be specified. If it is specified, program will read it but it will be ignored and replaced by the printed cell dimensions and angles in the restart PDB output file from GOMC (OutputName\_BOX\_0\_restart.pdb and OutputName\_BOX\_1\_restart.pdb).

**CBMC\_First** Number of CD-CBMC trials to choose the first atom position (Lennard-Jones trials for first seed growth).

- Value 1: Integer - Number of initial insertion sites to try.

**CBMC\_Nth** Number of CD-CBMC trials to choose the later atom positions (Lennard-Jones trials for first seed growth).

- Value 1: Integer - Number of LJ trials for growing later atom positions.

**CBMC\_Ang** Number of CD-CBMC bending angle trials to perform for geometry (per the coupled-decoupled CBMC scheme).

- Value 1: Integer - Number of trials per angle.

**CBMC\_Dih** Number of CD-CBMC dihedral angle trials to perform for geometry (per the coupled-decoupled CBMC scheme).

- Value 1: Integer - Number of trials per dihedral.

```
#####
# CBMC TRIALS
#####
CBMC_First  10
CBMC_Nth    4
CBMC_Ang    100
CBMC_Dih    30
```

Next section specifies the parameters that will be used for free energy calculation in NVT and NPT ensembles.

**FreeEnergyCalc** For NVT and NPT ensemble only: Considers to calculate the free energy data (the energy different between current lambda state and all other neighboring lambda states, and calculate the derivative of energy with respect to current lambda) or not. If it is set to true, the frequency of free energy calculation need to be set. The free energy data will be printed into Free\_Energy\_BOX\_0\_OutputName.dat.

- Value 1: Boolean - True enabling free energy calculation during the simulation, false disabling the calculation.
- Value 2: Ulong - The frequency of calculating the free energy.

**MoleculeType** Sets the solute molecule kind (residue name) and molecule number (residue ID), which absolute solvation free will be calculated for.

- Value 1: String - The solute name (residue name).

- Value 2: Integer - The solute molecule number (residue ID).

**InitialState** Sets the index of the `LambdaCoulomb` and `LambdaVDW` vectors, to determine the simulation lambda value for VDW and Coulomb interactions.

- Value 1: Integer - The index of `LambdaCoulomb` and `LambdaVDW` vectors.

**LambdaVDW** Sets the intermediate lambda states to which solute-solvent VDW interaction to be scaled.

- Value 1: Double - Lambda values for VDW interaction in ascending order.

**Warning:** All lambda values must be stated in the ascending order, otherwise the program will terminate.

**LambdaCoulomb** Sets the intermediate lambda states to which solute-solvent Coulombic interaction to be scaled.

- Value 1: Double - Lambda values for Coulombic interaction in ascending order.

**Warning:** All lambda values must be stated in the ascending order, otherwise the program will terminate.

---

**Note:**

- By default, the lambda values for Coulombic interaction will be set to zero if `ElectroStatic` or `Ewald` is **deactivated**.
  - By default, the lambda values for Coulombic interaction will be set to Lambda values for VDW interaction if `ElectroStatic` or `Ewald` is **activated**.
- 

**ScaleCoulomb** Determines to scale the Coulombic interaction non-linearly (soft-core scheme) or not.

- Value 1: Boolean - True if coulombic interaction needs to be scaled non-linearly, False if coulombic interaction needs to be scaled linearly.

---

**Note:** By default, the `ScaleCoulomb` will be set to false.

---

**ScalePower** Sets the  $p$  value in soft-core scaling scheme, where the distance between solute and solvent is scaled non-linearly.

- Value 1: Integer - The  $p$  value in the soft-core scaling scheme.

---

**Note:** By default, the `ScalePower` will be set to 2.

---

**ScaleAlpha** Sets the  $\alpha$  value in soft-core scaling scheme, where the distance between solute and solvent is scaled non-linearly.

- Value 1: Double -  $\alpha$  value in the soft-core scaling scheme.

---

**Note:** By default, the `ScaleAlpha` will be set to 0.5.

---

**MinSigma** Sets the minimum  $\sigma$  value in soft-core scaling scheme, where the distance between solute and solvent is scaled non-linearly.

- Value 1: Double - Minimum  $\sigma$  value in the soft-core scaling scheme.

---

**Note:** By default, the MinSigma will be set to 3.0.

---



---

**Note:** Scaling the distance between solute and solvent using soft-core scheme:

$$r_{sc} = \left[ \alpha (1 - \lambda)^p \sigma^6 + r^6 \right]^{\frac{1}{6}}$$


---

Here is the example of solvation free energy of CO2, in intermediate state 3.

```
#####
# FREE ENERGY PARAMETERS
#####
FreeEnergyCalc true    1000
MoleculeType  CO2    1
InitialState   3
ScalePower     2
ScaleAlpha     0.5
MinSigma       3.0
ScaleCoulomb   false
#states        0      1      2      3      4
LambdaVDW      0.00  0.50  1.00  1.00  1.00
LambdaCoulomb  0.00  0.00  0.00  0.50  1.00
```

## 9.10.2 Output Controls

This section contains all the values that control output in the control file. For example, certain variables control the naming of files outputted of the block-averaged thermodynamic variables of interest, the PDB files, etc.

**OutputName** Unique name with no space for simulation used to name the block average, PDB, and PSF output files.

- Value 1: String - Unique phrase to identify this system.

```
#####
# OUTPUT FILE NAME
#####
OutputName  ISB_T_270_K
```

**CoordinatesFreq** Controls output of PDB file (coordinates). If PDB outputting was enabled, one file for NVT or NPT and two files for Gibbs ensemble or GC ensemble will be outputted into OutputName\_BOX\_n.pdb, where n defines the box number.

- Value 1: Boolean - “true” enables outputting these files; “false” disables outputting.
- Value 2: Ulong - Steps per dump PDB frame. It should be less than or equal to RunSteps. If this keyword could not be found in configuration file, its value will be assigned a default value to dump 10 frames.

---

**Note:**

- DCDFreq should be used unless the low precision and slower PDB trajectory is needed, perhaps beta and occupancy values are desired.

- The PDB file contains an entry for every ATOM, in all boxes read. This allows VMD (which requires a constant number of atoms) to properly parse frames, with a bit of help. Atoms that are not currently in a specific box are given the coordinate (0.00, 0.00, 0.00). The occupancy value corresponds to the box a molecule is currently in (e.g. 0.00 for box 0; 1.00 for box 1).
  - At the beginning of simulation, a merged PSF file will be outputted into `OutputName_merged.psf`, in which all boxes will be outputted. It also contains the topology for every molecule in both boxes, corresponding to the merged PDB format. Loading PDB files into merged PSF file in VMD allows the user to visualize and analyze the results.
- 

**DCDFreq** Controls output of DCD file (binary coordinates). If DCD outputting was enabled, one file for NVT or NPT and two files for Gibbs ensemble or GC ensemble will be outputted into `OutputName_BOX_n.dcd`, where n defines the box number.

- Value 1: Boolean - “true” enables outputting these files; “false” disables outputting.
  - Value 2: Ulong - Steps per dump PDB frame. It should be less than or equal to `RunSteps`. If this keyword could not be found in configuration file, its value will be assigned a default value to dump 10 frames.
- 

**Note:**

- The DCD file contains an entry for every ATOM, in all boxes read. This allows VMD (which requires a constant number of atoms) to properly parse frames, with a bit of help. Atoms that are not currently in a specific box are given the coordinate (0.00, 0.00, 0.00). The occupancy value corresponds to the box a molecule is currently in (e.g. 0.00 for box 0; 1.00 for box 1).
  - At the beginning of simulation, a merged PSF file will be outputted into `OutputName_merged.psf`, in which all boxes will be outputted. It also contains the topology for every molecule in both boxes, corresponding to the merged PDB format. Loading DCD files into merged PSF file in VMD allows the user to visualize and analyze the results.
- 

**RestartFreq** Controls the output of the last state of simulation at a specified step in

- PDB files (coordinates)
- PSF files (structure)
- XSC files (box dimensions)
- COOR files (binary coordinates)
- CHK files (checkpoint)
- If provided as input: VEL files (velocity)

`OutputName_BOX_n_restart.*`, where n defines the box number. Header part of this file contains important information and will be needed to restart the simulation:

Restart PDB files, one file for NVT or NPT and two files for Gibbs ensemble or GC ensemble, will be outputted with the following information. - Simulation cell dimensions and angles.

- Maximum amount of displacement ( $\text{\AA}$ ), rotation ( $\delta$ ), and volume ( $\text{\AA}^3$ ) that used in Displacement, Rotation, and Volume move.
- 

**Note:**

- The restart PDB/PSF/COOR/VEL files contains only ATOM that exist in each boxes at specified steps. These box restart files allows the user to load a box into NAMD and run molecular dynamics in Hybrid Monte-Carlo Molecular Dynamics (py-MCMD).

- When restarting the GOMC simulation from two restart files, the order of the molecules in the trajectory may differ preventing trajectory concatenation, unless the CHK file is loaded.
  - Only restart files must be used to begin a GOMC simulation with `Restart` simulation active. The merged psf is NOT a restart file.
  - `CoordinatesFreq` must be a common multiple of `RestartFreq` or vice versa.
- 

**ConsoleFreq** Controls the output to STDIO (“the console”) of messages such as acceptance statistics, and run timing info. In addition, instantaneously-selected thermodynamic properties will be output to this file.

- Value 1: Boolean - “true” enables message printing; “false” disables outputting.
- Value 2: Ulong - Number of steps per print. If this keyword could not be found in the configuration file, the value will be assigned by default to dump 1000 output for `RunSteps` greater than 1000 steps and 100 output for `RunSteps` less than 1000 steps.

**BlockAverageFreq** Controls the block averages output of selected thermodynamic properties. Block averages are averages of thermodynamic values of interest for chunks of the simulation (for post-processing of averages or std. dev. in those values).

- Value 1: Boolean - “true” enables printing block average; “false” disables it.
- Value 2: Ulong - Number of steps per block-average output file. If this keyword cannot be found in the configuration file, its value will be assigned a default to dump 100 output.

**HistogramFreq** Controls the histograms. Histograms are a binned listing of observation frequency for a specific thermodynamic variable. In this code, they also control the output of a file containing energy/molecule samples; it only will be used in GC ensemble simulations for histogram reweighting purposes.

- Value 1: Boolean - “true” enables printing histogram; “false” disables it.
- Value 2: Ulong - Number of steps per histogram output file. If this keyword cannot be found in the configuration file, a value will be assigned by default to dump 1000 output for `RunSteps` greater than 1000 steps and 100 output for `RunSteps` less than 1000 steps.

```
#####  
# STATISTICS Enable, Freq.  
#####  
CoordinatesFreq   true 10000000  
RestartFreq       true 1000000  
CheckpointFreq    true 1000000  
ConsoleFreq       true 100000  
BlockAverageFreq  true 100000  
HistogramFreq     true 10000
```

The next section controls the output of the energy/molecule sample file and the distribution file for molecule counts, commonly referred to as the “histogram” output. This section is only required if Grand Canonical ensemble simulation was used.

**DistName** Sets short phrase to naming molecule distribution file.

- Value 1: String - Short phrase which will be combined with *RunNumber* and *RunLetter* to use in the name of the binned histogram for molecule distribution.

**HistName** Sets short phrase to naming energy sample file.

- Value 1: String - Short phrase, which will be combined with *RunNumber* and *RunLetter*, to use in the name of the energy/molecule count sample file.

**RunNumber** Sets a number, which is a part of *DistName* and *HistName* file name.



- Value 1: Uint – Run number to be used in the above file names.

**RunLetter** Sets a letter, which is a part of *DistName* and *HistName* file name.

- Value 1: Character – Run letter to be used in above file names.

**SampleFreq** Controls histogram sampling frequency.

- Value 1: Uint – the number of steps per histogram sample.

```
#####
# OutHistSettings
#####
DistName  dis
HistName  his
RunNumber 5
RunLetter a
SampleFreq 200
```

**OutEnergy, OutPressure, OutMolNumber, OutDensity, OutVolume, OutSurfaceTension** Enables/Disables for specific kinds of file output for tracked thermodynamic quantities

- Value 1: Boolean – “true” enables message output of block averages via this tracked parameter (and in some cases such as entry, components); “false” disables it.
- Value 2: Boolean – “true” enables message output of a fluctuation into the console file via this tracked parameter (and in some cases, such as entry, components); “false” disables it.

The keywords are available for the following ensembles

Keyword	NVT	NPT & Gibbs	GC
OutEnergy	✓	✓	✓
OutPressure	✓	✓	✓
OutMolNumber		✓	✓
OutDensity		✓	✓
OutVolume		✓	✓
OutSurfaceTension	✓		

Here is an example:

```
#####
# ENABLE: BLK AVE., FLUC.
#####
OutEnergy      true true
OutPressure     true true
OutMolNum       true true
OutDensity      true true
OutVolume       true true
OutSurfaceTention false false
```



## GOMC'S OUTPUT FILES

**GOMC currently supports several kinds of output:**

- STDIO (“console”) output
- File output
  - Block Averages
  - PDB
  - PSF
  - Molecule distribution (GCMC only)
  - Histogram (GCMC only)
  - Free energy data (NVT and NPT only)

GOMC output units:

Properties	Units
Energy	$K$
Pressure, Pressure Tensor	bar
Heat of vaporization	$KJ/mol$
Volume	$^3$
Density	$kg/m^3$
Mol Density	$molecule/^3$
Surface Tension	$mN/m$
Free Energy	$KJ/mol$

### 10.1 Console Output

A variety of useful information relating to instantaneous statistical and thermodynamic data (move trials, acceptance rates, file I/O messages warnings, and other kinds of information) is printed to the STDIO, which, in Linux, will typically be displayed in the terminal. This output can be redirected into a log file in Linux using the > operator.

```
$ GOMC CPU NVT in.conf > out_isobutane.log &
```

Statistical and thermodynamic information is provided in console output.

- Energy
  - Intermolecular (LJ)
  - Intramolecular bonded

- Intramolecular nonbonded
- Tail corrections
- Electrostatic real
- Electrostatic Reciprocal
- Electrostatic self
- Electrostatic correction
- Total electrostatic energy (sum of real, reciprocal, self, and correction)
- Total Energy (sum of the all energies)
- Pressure
  - Pressure Tensor ( $P_{xx}, P_{yy}, P_{zz}$ )
  - Total pressure
- Statistic
  - Volume
  - Pressure
  - Total molecule number
  - Total Density
  - Surface Tension
  - Mol fraction of each species
  - Mol density of each species

Detailed move, energy, and statistical or thermodynamic information for each simulation box will be printed in three different sections. Each section's title will start with MTITLE, ETITLE, and STITLE for move, energy, and statistical information, respectively. The instantaneous values for each section will start with MOVE\_#, ENER\_#, and STAT\_# for move, energy, and statistical values, respectively. Where, # is the simulation box number. In addition, if pressure calculation is activated and enabled to print, pressure tensor will be printed in the console output file. This section starts with PRES\_# and print the diagonal value of pressure tensor  $P_{xx}$ ,  $P_{yy}$ , and  $P_{zz}$ , respectively. The second element after the title of each section is the step number.

In order to extract the desired information from the console file, “grep” and “awk” commands can be used with a proper title section. For example, in order to extract total energy of the system, the following command needs to be executed in terminal:

```
$ grep "ENER_0" output_console.log | awk '{print $3}'
```

Here, “output\_console.log” is the console output file and “\$3” represents the second element of the “ENERGY\_BOX\_0” section.

---

**Note:** Surface Tension is calculated using Virial method according to following equation,

$$\gamma = \frac{1}{2A_{xy}} \int_0^L \left( P_{zz} - \frac{P_{xx} + P_{yy}}{2} \right) dz$$

---

The first section of this console output typically includes some information relating the system, CPU, GPU, and RAM. In continue, console output includes information regarding the input file (configuration file), force field reading, summary of the structure of the molecule, bonded and non-bonded parameters, and minimum and maximum coordinate

of molecules. This output is important; it may contain text relating to issues encountered if there was an error in the current run (e.g. a bad parameter, unknown keyword, missing parameters in the configuration file, etc.)

Next, the energy and statistic title, initial energy and statistic of the system's starting configuration will print:

---

**Note:** The frequency of printing MOVE\_#, ENER\_#, STAT\_#, and PRES\_# is controlled by ConsoleFreq parameter in configuration file.

---



---

**Note:** User can control the output of the thermodynamic properties in ENER\_# and STAT\_# using the following parameters in configuration file:

---

```
#####
# ENABLE:          BLK, FLUC.
#####
OutEnergy          true  true
OutPressure         true  true
OutMolNum           true  true
OutDensity          true  true
OutVolume           true  true
OutSurfaceTention  false false
```

---



---

**Note:** If total energy of simulation is greater than  $1.0e^{12}$ , System Total Energy Calculation will be performed at EqSteps to preserve energy value.

---

After the simulation starts, move, energy, and statistical title, followed by their values for each simulation box, will print:

At the end of the run, Monte Carlo move acceptance for each molecule kind and simulation box, total amount of time spent on each Monte Carlo move, total timing information, and other wrap up info will be printed.

---

**Note:**

- Printed energy and statistical values are instantaneous values.
  - In order to keep the format of console file consistent and print the calculated properties with high accuracy, scientific format is used.
  - It's important to watch the acceptance rates and adjust the move percentages and CBMC trial amounts to get the desired rate of move acceptance.
- 

## 10.2 Block Output Files

GOMC tracks a number of thermodynamic variables of interest during the simulation and prints them all in one file for each box.

- Energy
  - Intermolecular (LJ)
  - Intramolecular bonded

```
Info: GOMC Version 2.40
Info: Start Time: Mon Mar  4 14:40:36 2019
Info: Host Name: #####
CPU information:
Info: Total number of CPUs: 6
Info: Total number of CPUs available: 6
Info: Model name: Intel(R) Core(TM) i5-8600K CPU @ 3.60GHz
Info: System name: Linux
Info: Release: 3.10.0-957.1.3.el7.x86_64
Info: Version: #1 SMP Mon Nov 26 12:36:06 CST 2018
Info: Kernel Architecture: x86_64
Info: Total Ram: 15840.4MB
Info: Used Ram: 9135.3MB
Info: Working in the current directory: /soroush/GOMC_Examples/GCMC/isobutane/run2a_bridge
Info: GOMC COMPILED TO RUN GRAND CANONICAL ENSEMBLE.
Info: Number of threads          1

Reading Input File:              in.conf
Info: Random seed                Active
Info: PARAMETER file             CHARMM format!
Info: Input Temperature          410.0000 K
Info: Non-truncated potential    Active
Info: Long Range Correction      Active
Info: Cutoff                    10.0000 A
Info: Exclude                   ONE-FOUR
Info: Pressure calculation       Inactive
Info: Total number of steps      1000000
Info: Number of equilibration steps 500000
Info: Move adjustment frequency  1000
Info: Displacement move frequency 0.2000
Info: Rotation move frequency    0.1000
Info: Molecule swap move frequency 0.6000
Info: Regrowth move frequency    0.1000
Info: CBMC First atom trials     10
Info: CBMC Secondary atom trials 4
Info: CBMC Angle trials         100
Info: CBMC Dihedral trials       30
Info: Chemical potential         ISB      -3135.0000 K
Info: Output name                ISB_410_00_K_u_3135_r1a
Info: Coordinate frequency       1000000
Info: Restart frequency          1000000
Info: Console output frequency   10000
Info: Average output frequency   100000
Info: Histogram output frequency 100000
Info: Histogram sample frequency 200
Default: Intra-Swap move frequency 0.0000
Default: Intra-MEMC move frequency 0.0000
Default: Crank-Shaft move frequency 0.0000
Default: MEMC move frequency     0.0000
Default: Short Range Cutoff      0.0000
Warning: 1-4 Electrostatic scaling set, but will be ignored.
Finished Reading Input File:      in.conf
```

Fig. 1: Printing summary of configuration file.

```

Reading from CHARMM-Style parameter file:      ../../../../common/Par_TraPPE_Alkanes_CHARMM.inp
Reading BONDS parameters.
Reading ANGLES parameters.
Reading DIHEDRALS parameters.
Reading NONBONDED parameters.
Finished reading CHARMM-Style parameter file:  ../../../../common/Par_TraPPE_Alkanes_CHARMM.inp
Reading from box 0 PDB coordinate file:        STEP3_START_ISB_vap_BOX_0.pdb
Finished reading box 0 PDB coordinate file:    STEP3_START_ISB_vap_BOX_0.pdb
Reading from box 1 PDB coordinate file:        STEP3_START_ISB_reservoir_BOX_1.pdb
Finished reading box 1 PDB coordinate file:    STEP3_START_ISB_reservoir_BOX_1.pdb
Random number seed: 2984217758

```

Molecules in PSF:

Molecule Kind: ISB

Idx	name	type	charge	mass
0	C1	CH1	0.0000	13.0190
1	C2	CH3	0.0000	15.0350
2	C3	CH3	0.0000	15.0350
3	C4	CH3	0.0000	15.0350

Bonds:

[0 1] [0 2] [0 3]

Angles:

[1 0 3] [1 0 2] [2 0 3]

Dihedrals:

Bonds parameter:

Atom Types	Kb(K)	b0(A)
CH1 CH3	FIX	1.5400

Angles parameter:

Atom Types	Ktheta(K)	theta0(degree)
CH3 CH1 CH3	31250.0025	112.0000

Dihedrals parameter:

Atom Types	Kchi(K)	n	delta(degree)
------------	---------	---	---------------

NonBonded 1-4 parameters:

Type1	Type2	Epsilon(K)	Sigma(A)	N
CH1	CH1	10.0000	4.6800	12.00
CH1	CH3	31.3050	4.2150	12.00
CH3	CH3	98.0000	3.7500	12.00

NonBonded parameters:

Type1	Type2	Epsilon(K)	Sigma(A)	N
CH1	CH1	10.0000	4.6800	12.00
CH1	CH3	31.3050	4.2150	12.00
CH3	CH3	98.0000	3.7500	12.00

Fig. 2: Reading parameter file and printing the summary of the force field.

```

Info: Box 0: Periodic Cell Basis 1 30.000 0.000 0.000
Info: Box 0: Periodic Cell Basis 2 0.000 30.000 0.000
Info: Box 0: Periodic Cell Basis 3 0.000 0.000 30.000

Info: Box 1: Periodic Cell Basis 1 30.000 0.000 0.000
Info: Box 1: Periodic Cell Basis 2 0.000 30.000 0.000
Info: Box 1: Periodic Cell Basis 3 0.000 0.000 30.000

Minimum coordinates in box 0: x = 1.000, y = 20.816, z = 25.283
Maximum coordinates in box 0: x = 5.215, y = 28.682, z = 29.000
Wrapping molecules inside the simulation box 0:
Minimum coordinates in box 1: x = 1.000, y = 1.000, z = 1.000
Maximum coordinates in box 1: x = 29.000, y = 29.000, z = 29.000
Wrapping molecules inside the simulation box 1:

```

Fig. 3: Reading the PDB files for each box, printing the min and max coordinates.

```

#####
##### INITIAL SIMULATION ENERGY #####
#####

ETITLE:      STEP      TOTAL      INTRA(B)      INTRA(NB)      INTER(LJ)
           LRC      TOTAL_ELECT      REAL      RECIP      SELF
           CORR

ENER_0:       0      2.0645e+02      3.6445e+02      0.0000e+00      -1.5353e+02
           -4.4761e+00      0.0000e+00      0.0000e+00      0.0000e+00      0.0000e+00
           -0.0000e+00

ENER_1:       0      1.0969e+05      1.0969e+05      0.0000e+00      0.0000e+00
           0.0000e+00      0.0000e+00      0.0000e+00      0.0000e+00      0.0000e+00
           -0.0000e+00

STITLE:      STEP      TOTALMOL      TOT_DENSITY
STAT_0:       0           2      7.1494e+00
STAT_1:       0          600      2.1448e+03

```

Fig. 4: Printing initial energy of the system and statistical values.



```
#####
##### STARTING SIMULATION #####
```

MTITLE:	STEP	DISTRY	DISACCEPT	DISACCEPT%	DISMAX
	ROTATE	ROTACCEPT	ROTACCEPT%	ROTMAX	REGROWTH
	REGROWACCEPT	REGROWACCEPT%	TRANSFER	TRANACCEPT	TRANACCEPT%

ETITLE:	STEP	TOTAL	INTRA(B)	INTRA(NB)	INTER(LJ)
	LRC	TOTAL_ELECT	REAL	RECIP	SELF
	CORR				

STITLE:	STEP	TOTALMOL	TOT_DENSITY
---------	------	----------	-------------

Printed combined psf to file ISB\_410\_00\_K\_u\_3135\_r1a\_merged.psf

MOVE_0:	10000	1980	862	4.3535e+01	1.4689e+00
	997	781	7.8335e+01	3.1416e+00	1048
	668	6.3740e+01	3051	1609	5.2737e+01

ENER_0:	10000	-1.7736e+04	4.1283e+04	0.0000e+00	-5.3378e+04
	-5.6410e+03	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	0.0000e+00				

STAT_0:	10000	71	2.5380e+02
---------	-------	----	------------

MOVE_1:	10000	2924	1540	5.2668e+01
---------	-------	------	------	------------

ENER_1:	10000	3.3227e+05	3.3227e+05	0.0000e+00	0.0000e+00
	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	0.0000e+00				

STAT_1:	10000	531	1.8982e+03
---------	-------	-----	------------

Steps/sec: 5470.897, Simulation ends in: 0 d: 0 h: 3 m

- Intramolecular nonbonded
- Tail corrections
- Electrostatic real
- Electrostatic Reciprocal
- Total Energy (sum of the all energies)
- Virial
- Statistic
  - Pressure
  - Surface Tension (using virial method)
  - Volume
  - Total molecule number
  - Total Density
  - Mol fraction of each species
  - Mol density of each species
  - Heat of vaporization

```

#STEPS          TOT_EN          EN_INTER          EN_TC          EN_INTRA(B)
EN_INTRA(NB)    EN_ELECT        EN_REAL          EN_RECIP        TOTAL_VIR
PRESSURE        TOT_MOL          TOT_DENS
100000          4.00827177e+05 -4.31993824e+04 -3.00811270e+03 4.47034672e+05
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 8.28858977e+04
4.50352244e+00 1.00000000e+03 9.60930208e+00

```

Fig. 5: Printing the average energy of the system and statistical values.

At the beginning of each file, the title of each property followed by their average values is printed. Desired data can be extracted, as explained before, using the “awk” command. For example, in order to extract total density of the system, the following command need to be executed in terminal:

```
$ cat Blk_OutputName_BOX_0.dat | awk '{print $2}'
```

Here, “Blk\_OutputName\_BOX\_0.dat” is the block-average file for simulation box 0 and “\$2” represents the second column of the block file.

**Note:** The frequency of printing average thermodynamic properties is controlled by `BlockAverageFreq` parameter in configuration file.

**Note:** User can control the output of the average thermodynamic properties, using the following parameters in configuration file:

```

#####
# ENABLE:          BLK, FLUC.
#####
OutEnergy          true true

```

(continues on next page)

(continued from previous page)

OutPressure	true	true
OutMolNum	true	true
OutDensity	true	true
OutVolume	true	true
OutSurfaceTention	false	false

---

**Note:** In order to keep the format of BlockOutput file consistent and print the calculated properties with high accuracy, scientific format is used.

---

## 10.3 PDB Output Files

GOMC is capable of outputting the molecular coordinates during the simulation in PDB format. GOMC outputs two types of PDB files:

1. The last state of simulation at a specified step (OutputName\_BOX\_n.pdb, where n defines the box number).
2. The state of simulation at a specified step (OutputName\_BOX\_n\_restart.pdb, where n defines the box number).

### 10.3.1 1. Restart Trajectory

The restart PDB file contains only ATOM that exist in each box at specified steps. This allows the user to load this file into GOMC once Restart simulation was active. If restart PDB output was enabled, one file for NVT or NPT and two files for Gibbs ensemble or grand canonical ensemble will be outputted. Header part of this file contains important information and will be needed to restart the simulation:

- Simulation cell dimensions and angles.
- Maximum amount of displacement ( $\text{\AA}$ ), rotation ( $\delta$ ), and volume ( $^3$ ) that used in Displacement, Rotation, and Volume move.

---

**Note:** The frequency of printing restart PDB file is controlled by RestartFreq parameter in configuration file.

---



---

**Important:** The beta value in restart PDB file defines the mobility of the molecule.

- Beta = 0.00: molecule can move and transfer within and between boxes.
  - Beta = 1.00: molecule is fixed in its position.
  - Beta = 2.00: molecule can move within the box but cannot be transferred between boxes.
-

```

REMARK      GOMC      107.880      3.14159      500.000      30000
CRYST1 215.760 215.760 215.760 90.00 90.00 90.00 P 1 1
ATOM      1 C1  ISB A  1      91.014 189.172 16.799 0.00 0.00
ATOM      2 C2  ISB A  1      91.356 187.671 16.792 0.00 0.00
ATOM      3 C3  ISB A  1      91.502 189.816 15.487 0.00 0.00
ATOM      4 C4  ISB A  1      91.708 189.852 17.993 0.00 0.00
ATOM      5 C1  ISB A  2     126.989 101.879 104.481 0.00 0.00
ATOM      6 C2  ISB A  2     128.341 101.186 104.733 0.00 0.00
ATOM      7 C3  ISB A  2     127.229 103.358 104.125 0.00 0.00
ATOM      8 C4  ISB A  2     126.120 101.791 105.749 0.00 0.00
ATOM      9 C1  ISB A  3      46.228 110.033 173.720 0.00 0.00
ATOM     10 C2  ISB A  3      47.417 109.105 173.416 0.00 0.00
ATOM     11 C3  ISB A  3      45.724 109.769 175.151 0.00 0.00
ATOM     12 C4  ISB A  3      45.092 109.760 172.717 0.00 0.00
ATOM     13 C1  ISB A  4      93.796  95.717  78.101 0.00 0.00
ATOM     14 C2  ISB A  4      94.128  97.060  78.777 0.00 0.00
ATOM     15 C3  ISB A  4      94.024  95.835  76.583 0.00 0.00
ATOM     16 C4  ISB A  4      92.324  95.355  78.372 0.00 0.00

```

Fig. 6: The coordinates of isobutane molecules in simulation Box 0, at steps 30000, in OutputName\_BOX\_0\_restart.pdb file.

```

REMARK      GOMC      60.000      3.14159      500.000      30000
CRYST1 120.000 120.000 120.000 90.00 90.00 90.00 P 1 1
ATOM      1 C1  ISB A  1      39.361   3.563 108.962 0.00 0.00
ATOM      2 C2  ISB A  1      39.132   2.257 109.744 0.00 0.00
ATOM      3 C3  ISB A  1      38.718   3.540 107.563 0.00 0.00
ATOM      4 C4  ISB A  1      39.012   4.795 109.818 0.00 0.00
ATOM      5 C1  ISB A  2      86.302  31.431  33.813 0.00 0.00
ATOM      6 C2  ISB A  2      87.270  31.227  32.633 0.00 0.00
ATOM      7 C3  ISB A  2      85.662  30.135  34.344 0.00 0.00
ATOM      8 C4  ISB A  2      86.906  32.266  34.958 0.00 0.00
ATOM      9 C1  ISB A  3      44.228  31.138  85.407 0.00 0.00
ATOM     10 C2  ISB A  3      45.731  31.164  85.073 0.00 0.00
ATOM     11 C3  ISB A  3      43.531  29.891  84.833 0.00 0.00
ATOM     12 C4  ISB A  3      43.451  32.366  84.896 0.00 0.00
ATOM     13 C1  ISB A  4       5.564 100.487  58.187 0.00 0.00
ATOM     14 C2  ISB A  4       5.595 101.260  59.518 0.00 0.00
ATOM     15 C3  ISB A  4       4.375  99.509  58.223 0.00 0.00
ATOM     16 C4  ISB A  4       6.901  99.765  57.934 0.00 0.00

```

Fig. 7: The coordinates of isobutane molecules in simulation Box 1, at steps 30000, in OutputName\_BOX\_1\_restart.pdb file.

### 10.3.2 2. Simulation Trajectories

The trajectory PDB file contains an entry for every ATOM, in all boxes read. This allows VMD (which requires a constant number of atoms) to properly parse the simulation frames. If PDB output was enabled, one file for NVT or NPT and two files for Gibbs ensemble or grand canonical ensemble will be outputted. Header part of this file contains simulation cell dimensions and angles, frame number, and simulation steps.

```
REMARK      GOMC      1
CRYST1 215.760 215.760 215.760 90.00 90.00 90.00 P 1 1
ATOM      1 C1  ISB A  1      72.840 114.295 143.266 0.00 0.00
ATOM      2 C2  ISB A  1      71.755 113.691 142.355 0.00 0.00
ATOM      3 C3  ISB A  1      72.864 115.825 143.090 0.00 0.00
ATOM      4 C4  ISB A  1      74.213 113.713 142.886 0.00 0.00
ATOM      5 C1  ISB A  2      53.884  92.769 180.599 0.00 0.00
ATOM      6 C2  ISB A  2      54.024  91.247 180.788 0.00 0.00
ATOM      7 C3  ISB A  2      52.398  93.127 180.408 0.00 0.00
ATOM      8 C4  ISB A  2      54.681  93.209 179.358 0.00 0.00
ATOM      9 C1  ISB A  3      56.176  39.274 137.006 0.00 0.00
ATOM     10 C2  ISB A  3      57.525  38.577 136.755 0.00 0.00
ATOM     11 C3  ISB A  3      56.349  40.797 136.857 0.00 0.00
ATOM     12 C4  ISB A  3      55.687  38.950 138.430 0.00 0.00
ATOM     13 C1  ISB A  4     129.036 119.026  57.538 0.00 0.00
ATOM     14 C2  ISB A  4     130.277 118.344  56.934 0.00 0.00
ATOM     15 C3  ISB A  4     127.868 118.023  57.579 0.00 0.00
ATOM     16 C4  ISB A  4     128.642 120.236  56.671 0.00 0.00
```

Fig. 8: The coordinates of all isobutane molecules at beginning of the simulation, in OutputName\_BOX\_0.pdb file.

```
REMARK      GOMC      1
CRYST1 120.000 120.000 120.000 90.00 90.00 90.00 P 1 1
ATOM      1 C1  ISB A  1      0.000  0.000  0.000 0.00 0.00
ATOM      2 C2  ISB A  1      0.000  0.000  0.000 0.00 0.00
ATOM      3 C3  ISB A  1      0.000  0.000  0.000 0.00 0.00
ATOM      4 C4  ISB A  1      0.000  0.000  0.000 0.00 0.00
ATOM      5 C1  ISB A  2      0.000  0.000  0.000 0.00 0.00
ATOM      6 C2  ISB A  2      0.000  0.000  0.000 0.00 0.00
ATOM      7 C3  ISB A  2      0.000  0.000  0.000 0.00 0.00
ATOM      8 C4  ISB A  2      0.000  0.000  0.000 0.00 0.00
ATOM      9 C1  ISB A  3      0.000  0.000  0.000 0.00 0.00
ATOM     10 C2  ISB A  3      0.000  0.000  0.000 0.00 0.00
ATOM     11 C3  ISB A  3      0.000  0.000  0.000 0.00 0.00
ATOM     12 C4  ISB A  3      0.000  0.000  0.000 0.00 0.00
ATOM     13 C1  ISB A  4      0.000  0.000  0.000 0.00 0.00
ATOM     14 C2  ISB A  4      0.000  0.000  0.000 0.00 0.00
ATOM     15 C3  ISB A  4      0.000  0.000  0.000 0.00 0.00
ATOM     16 C4  ISB A  4      0.000  0.000  0.000 0.00 0.00
```

Fig. 9: The coordinates of all isobutane molecules at beginning of the simulation, in OutputName\_BOX\_1.pdb file.

---

**Note:** The frequency of printing trajectory PDB file is controlled by CoordinatesFreq parameter in configuration file.

---



---

**Important:**

---

- For atoms not currently in a box, the coordinates are set to `< 0.00, 0.00, 0.00 >`.
  - The occupancy value defines the box, which molecule is in (box 0 occupancy=0.00 ; box 1 occupancy=1.00)
  - The beta value in trajectory PDB file defines the mobility of the molecule.
    - Beta = 0.00: molecule can move and transfer within and between boxes.
    - Beta = 1.00: molecule is fixed in its position.
    - Beta = 2.00: molecule can move within the box but cannot be transferred between boxes.
- 

## 10.4 PSF Output File

At the beginning of the simulation, a merged PSF file will be outputted into `OutputName_merged.psf`. It contains the topology for every molecule in all simulation boxes, corresponding to the merged PDB format. Loading PDB files into merged PSF file in VMD allows the user to visualize and analyze the results.

The PSF file contains six main sections: `remarks`, `atoms`, `bonds`, `angles`, and `dihedrals`. Each section starts with a specific header described below:

- `NTITLE`: Remarks on the file.

PSF

```
2 !NTITLE
REMARKS Combined PSF produced by GOMC
REMARKS Contains Geometry data for molecules in ALL boxes in the system
```

Fig. 10: Remaks generated by GOMC.

- `NATOM`: The atom names, residue name, atom types, and partial charges of each atom.

```
8000 !NATOM
  1 ISB  1  ISB  C1  CH1  0.000000  13.0190  0
  2 ISB  1  ISB  C2  CH3  0.000000  15.0350  0
  3 ISB  1  ISB  C3  CH3  0.000000  15.0350  0
  4 ISB  1  ISB  C4  CH3  0.000000  15.0350  0
  5 ISB  2  ISB  C1  CH1  0.000000  13.0190  0
  6 ISB  2  ISB  C2  CH3  0.000000  15.0350  0
  7 ISB  2  ISB  C3  CH3  0.000000  15.0350  0
  8 ISB  2  ISB  C4  CH3  0.000000  15.0350  0
  9 ISB  3  ISB  C1  CH1  0.000000  13.0190  0
 10 ISB  3  ISB  C2  CH3  0.000000  15.0350  0
 11 ISB  3  ISB  C3  CH3  0.000000  15.0350  0
 12 ISB  3  ISB  C4  CH3  0.000000  15.0350  0
 13 ISB  4  ISB  C1  CH1  0.000000  13.0190  0
 14 ISB  4  ISB  C2  CH3  0.000000  15.0350  0
 15 ISB  4  ISB  C3  CH3  0.000000  15.0350  0
 16 ISB  4  ISB  C4  CH3  0.000000  15.0350  0
```

Fig. 11: Atom section, taken from a merged PSF file for isobutane. The fields in the atom section, from left to right are atom ID, segment name, residue ID, residue name, atom name, atom type, charge, mass, and an unused 0.

- `NBOND`: The covalent bond section lists four pairs of atoms per line.

```

6000 !NBOND: bonds
  1      2      1      3      1      4      5      6
  5      7      5      8      9     10      9     11
  9     12     13     14     13     15     13     16
 17     18     17     19     17     20     21     22
 21     23     21     24     25     26     25     27
 25     28     29     30     29     31     29     32
 33     34     33     35     33     36     37     38
 37     39     37     40     41     42     41     43
 41     44     45     46     45     47     45     48
 49     50     49     51     49     52     53     54
 53     55     53     56     57     58     57     59

```

Fig. 12: Bond section, taken from a merged PSF file for isobutane.

- NTHETA: The angle section lists three triples of atoms per line.

```

6000 !NTHETA: angles
  2      1      4      2      1      3      3      1      4
  6      5      8      6      5      7      7      5      8
 10      9     12     10      9     11     11      9     12
 14     13     16     14     13     15     15     13     16
 18     17     20     18     17     19     19     17     20
 22     21     24     22     21     23     23     21     24
 26     25     28     26     25     27     27     25     28
 30     29     32     30     29     31     31     29     32
 34     33     36     34     33     35     35     33     36
 38     37     40     38     37     39     39     37     40
 42     41     44     42     41     43     43     41     44

```

Fig. 13: Angle section, taken from a merged PSF file for isobutane.

- NPHI: The dihedral sections list two quadruples of atoms per line.

```

0 !NPHI: dihedrals

```

Fig. 14: Dihedral section, taken from a merged PSF file for isobutane.

## 10.5 Molecule Distribution Output File

In grand canonical Monte Carlo (GCMC) simulation, GOMC outputs a binned number of molecules, observed in the system. This file can be used to detect the overlap between various GCMC simulation states. Sufficient overlap between various GCMC simulation is required in histogram reweighting method.

The molecule distribution will be outputted to a file, with a name constructed from parameters defined in configuration file (DistName, RunNumber, and RunLetter). For instance, for the first molecule kind and following parameters in configuration file

```
#####  
# OutHistSettings  
#####  
DistName    dis  
RunNumber    3  
RunLetter    a
```

GOMC will output the molecule distribution into “n1dis3a.dat” file.

---

**Important:** In case of system with multiple molecule kinds, multiple molecule distribution files will be outputted by GOMC (“n1dis3a.dat”, “n2dis3a.dat”, ...).

---

---

**Note:**

- The Molecule distribution files will be outputted at EqSteps.
  - The frequency of outputting molecule distribution file is controlled by HistogramFreq parameter in configuration file.
  - The observation frequency is controlled by SampleFreq parameter in configuration file.
- 

## 10.6 Histogram Output File

In grand canonical Monte Carlo (GCMC) simulation, GOMC outputs the observed number of molecule (for each molecule kind) and energy of the system (nonbonded + LRC). This file only will be used for histogram reweighting purposes.

The histogram will be outputted to a file, with a name constructed from parameters defined in configuration file (HistName, RunNumber, and RunLetter). For instance, for the following parameters in configuration file

```
#####  
# OutHistSettings  
#####  
HistName    his  
RunNumber    3  
RunLetter    a
```

GOMC will output the histogram data into “his3a.dat” file.

The header of the histogram file contains information of the simulated system, such as temperature, number of molecule kind, chemical potential, and x, y, z dimensions of simulation box.



```
96 1
97 1
98 1
99 3
100 1
101 1
103 9
104 8
105 15
106 23
107 25
108 31
109 39
110 35
111 60
112 58
113 76
114 78
115 87
116 94
117 111
118 104
119 126
120 108
121 136
122 127
123 137
124 132
125 131
126 137
127 107
128 89
129 90
130 70
131 69
132 53
133 33
134 39
135 21
136 14
137 7
138 4
139 3
140 1
142 1
144 3
145 1
```

Fig. 15: Molecule number distribution taken for isobutane simulation in GCMC simulation. The field in molecule distribution file, from left to right are number of molecule observed in the simulation and number of samples.

```
410 1 -3000 30 30 30
125      -187860.86069881968
128      -193131.7722388758
131      -197781.47348142008
127      -188820.92007212562
123      -175597.34923851598
124      -178872.94678179978
129      -199107.55645269612
127      -193257.69168095523
124      -184963.9123384784
124      -185202.98879885933
129      -198713.93659406601
124      -184408.40608603752
124      -184482.44070617665
124      -181990.23678435467
119      -165812.32245662552
109      -143632.61937537455
114      -157653.87157008398
114      -154524.87321516094
111      -147366.71317351784
109      -138669.84754439234
115      -152451.86525533127
108      -139624.95065114673
108      -137179.12801654899
111      -142586.4083805026
109      -141271.49002241078
109      -138438.29058796261
107      -137726.63617759256
111      -149700.20702121157
111      -149949.94194626925
110      -147502.12116691444
112      -153315.53586623361
105      -133608.00825840121
111      -139602.05471427058
109      -134575.28863465483
108      -134428.66996542862
112      -141081.35530106834
109      -138186.73387808813
107      -128985.59970412243
99       -120836.74857605236
106      -132341.92973198768
110      -139418.90375051263
109      -135460.094090267
116      -153090.94506563913
113      -142765.06317960389
112      -143204.33464886126
106      -130847.44605640478
106      -129854.94454478494
```

Fig. 16: The histogram taken for isobutane simulation in GCMC simulation. The field in histogram file, from left to right are number of molecule observed for the first molecule kind in the simulation and energy of the system (nonbonded + LRC).

---

**Important:** In case of system with multiple molecule kinds, multiple column will be printed, which each column represents the number of molecule for each molecule kind.

---

**Note:**

- The Histogram file will be outputted at EqSteps.
  - The frequency of outputting Histogram file is controlled by HistogramFreq parameter in configuration file.
  - The observation frequency is controlled by SampleFreq parameter in configuration file.
- 

## 10.7 Free Energy Output File

GOMC is capable of calculating absolute solvation free energy in NVT and NPT ensemble, using thermodynamic integration and free energy perturbation methods. GOMC outputs the raw informations, such as the derivative of energy with respect to current lambda ( $\frac{dE}{d\lambda}$ ) and energy different between current lambda state and all other neighboring lambda states ( $\Delta E_{\lambda_i \rightarrow \lambda_j}$ ), which is essential to calculate solvation free energy with various estimators, such as TI, BAR, MBAR, and more.

The header of Free\_Energy\_BOX\_0\_OutputName.dat contains the following information:

- Temperature of the simulation.
- The index of the lambda vector.
- Value of  $\lambda_{Coulomb}$  and  $\lambda_{VDW}$ .
- Monte Carlo step number.
- Total energy of the system.
- Derivative of energy with respect to lambda for coulomb interaction ( $\frac{dE}{d\lambda_{Coulomb}}$ ).
- Derivative of energy with respect to lambda for VDW interaction ( $\frac{dE}{d\lambda_{VDW}}$ ).
- Energy different between current lambda state and all other neighboring lambda states ( $\Delta E_{\lambda_i \rightarrow \lambda_j} = E_{\lambda_j} - E_{\lambda_i}$ )

```
#T = 298.0000(K), Lambda State 6: (lambda Coulomb, lambda VDW) = (0.5000,1.0000)
#Steps      Total_En(kJ/mol)      dU/dL(Coulomb=0.5000)      dU/dL(VDW=1.0000)  DelE(L->(0.00\
00,0.0000)) DelE(L->(0.0000,0.2500)) DelE(L->(0.0000,0.5000)) DelE(L->(0.0000,0.7500)) DelE(L->(0.00\
00,1.0000)) DelE(L->(0.2500,1.0000)) DelE(L->(0.5000,1.0000)) DelE(L->(0.7500,1.0000)) DelE(L->(1.00\
00,1.0000))
```

Fig. 17: Snapshot of GOMC free energy output file (Free\_Energy\_BOX\_0\_OutputName.dat).

---

**Important:** For simulation in NPT ensemble or NVT ensemble with activated pressure calculation (PressureCalc True), additional column will be printed to represent  $PV$  term.

---



---

**Note:** The frequency of outputting free energy data is controlled by FreeEnergyCalc parameter in configuration file.

---



## PUTTING IT ALL TOGETHER: RUNNING A GOMC SIMULATION

It is strongly recommended that you download the test system provided at [GOMC Website](#) or [Our Github Page](#)

Run different simulation types in order to become more familiar with different parameter and configuration files (\*.conf).

To recap the previous examples, a simulation of isobutane will be completed for a single temperature point on the saturated vapor-liquid coexistence curve.

The general plan for running the simulation is:

1. Build GOMC (if not done already)
2. Copy GOMC executable to build directory
3. Create scripts, PDB, and topology file to build the system, plus in.dat file and parameter files to prepare for runtime
4. Build finished PDBs and PSFs using the simulation.
5. Run the simulation in the terminal.
6. Analyze the output.

Please, complete steps 1 and 2; then, traverse to the directory, which should now contain a single file “GOMC\_CPU\_GEMC”. Next, six files need to be made:

- PDB file for isobutane
- Topology file describing isobutane residue
- Two \*.inp packmol scripts to pack two system boxes
- Two *TCL* scripts to input into PSFGen to generate the final configuration

isobutane.pdb

REMARK	1	File	created	by	GaussView	5.0.8			
ATOM	1	C1	ISB	1	0.911	-0.313	0.000	C	
ATOM	2	C2	ISB	1	1.424	-1.765	0.000	C	
ATOM	3	C3	ISB	1	-0.629	-0.313	0.000	C	
ATOM	4	C4	ISB	1	1.424	0.413	-1.257	C	
END									

Top\_Branched\_Alkane.inp

* Custom top file -- branched alkanes			
*			
MASS	1	CH3	15.035 C !

(continues on next page)

(continued from previous page)

```

MASS      2      CH1      13.019 C !

AUTOGENERATE ANGLES DIHEDRALS

RESI      ISB      0.00          ! isobutane { TraPPE }
GROUP
ATOM      C1      CH1      0.00    ! C3\
ATOM      C2      CH3      0.00    !   C2-C1
ATOM      C3      CH3      0.00    ! C4/
ATOM      C4      CH3      0.00    !
BOND      C1 C2 C1 C3 C1 C4
PATCHING FIRS NONE LAST NONE
END

```

pack\_box\_0.inp

```

tolerance  3.0
filetype    pdb
output      STEP2_ISB_packed_BOX_0.pdb

structure   isobutane.pdb
number      1000
inside cube 0. 0. 0. 68.00
end structure

```

pack\_box\_1.inp

```

tolerance  3.0
filetype    pdb
output      STEP2_ISB_packed_BOX_1.pdb

structure   isobutane.pdb
number      1000
inside cube 0. 0. 0. 68.00
end structure

```

build\_box\_0.inp

```

package require psfgen

topology ./Top Branched Alkane.inp segment ISB {
  pdb     ./STEP2_ISB_packed_BOX_0.pdb
  first   none
  last    none
}
coordpdb ./STEP2 ISB_packed_BOX_0.pdb ISB

writepsf ./STEP3_START_ISB_sys_BOX_0.psf
writepdb ./STEP3_START_ISB_sys_BOX_0.pdb

```

build\_box\_1.inp

```

package require psfgen

topology ./Top Branched Alkane.inp segment ISB {
  pdb    ./STEP2_ISB_packed_BOX_1.pdb
  first  none
  last   none
}
coordpdb ./STEP2 ISB_packed_BOX_1.pdb ISB

writepsf ./STEP3_START_ISB_sys_BOX_1.psf
writepdb ./STEP3_START_ISB_sys_BOX_1.pdb

```

These files can be created with a standard Linux or Windows text editor. Please, also copy a Packmol executable into the working directory.

Once those files are created, run in the terminal:

```

$ ./packmol < pack_box_0.inp
$ ./packmol < pack_box_1.inp

```

This will create the intermediate PDBs.

Then, run the PSFGen scripts to finish the system using the following commands:

```

$ vmd -dispdev text < ./build_box_0.inp
$ vmd -dispdev text < ./build_box_1.inp

```

This will create the intermediate PDBs.

To run the code a few additional things will be needed:

- A GOMC Gibbs ensemble executable
- A control file
- Parameter files

Enter the control file (in.conf) in the text editor in order to modify it. Example files for different simulation types can be found in previous section.

Once these four files have been added to the output directory, the simulation is ready.

Assuming the code is named GOMC\_CPU\_GEMC, run in the terminal using:

```

$ ./GOMC CPU GEMC in.conf > out_ISB_T_330.00_K_RUN_0.log &

```

For running GOMC in parallel, using openmp, run in the terminal using:

```

$ ./GOMC CPU GEMC +p4 in.conf > out_ISB_T_330.00_K_RUN_0.log&

```

Here, 4 defines the number of processors that will be used to run the simulation in parallel.

Progress can be monitored in the terminal with the tail command:

```

$ tail -f out_ISB.log

```

**Attention:** Congratulations! You have examined a single-phase coexistence point on the saturated vapor-liquid curve using GOMC operating in the Gibbs ensemble.

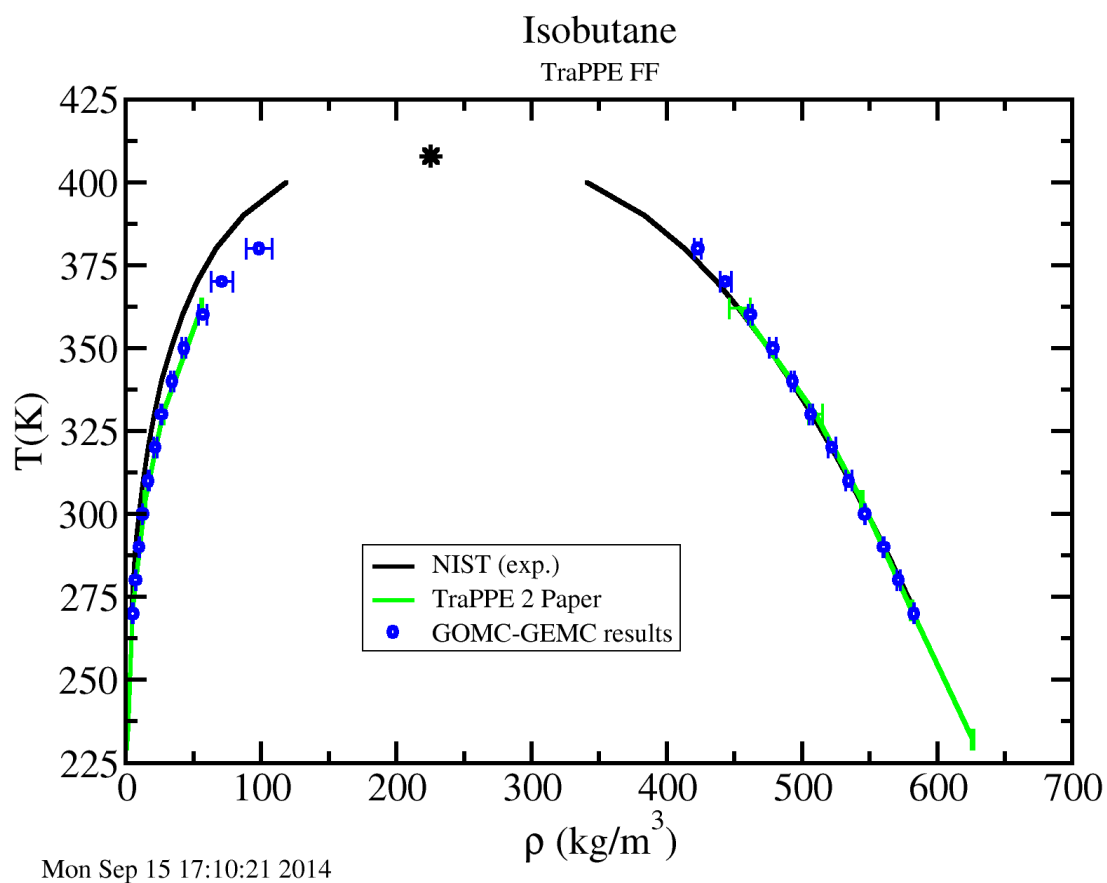


Fig. 1: Repeating this process for multiple temperatures will allow you to obtain the following results.



## INTERMOLECULAR ENERGY AND VIRIAL FUNCTION (VAN DER WAALS)

In this section, the virial and energy equation of Van der Waals interaction for different potential function are discussed in details.

### 12.1 VDW

This option calculates potential energy without any truncation.

**Potential Calculation** Interactions between atoms can be modeled with an n-6 potential, a Mie potential in which the attractive exponent is fixed. The Mie potential can be viewed as a generalized version of the 12-6 Lennard-Jones potential,

$$E_{VDW}(r_{ij}) = C_{n_{ij}} \epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $\sigma_{ij}$  are, respectively, the separation, minimum potential, and collision diameter for the pair of interaction sites  $i$  and  $j$ . The constant  $C_n$  is a normalization factor such that the minimum of the potential remains at  $-\epsilon_{ij}$  for all  $n_{ij}$ . In the 12-6 potential,  $C_n$  reduces to the familiar value of 4.

$$C_{n_{ij}} = \left( \frac{n_{ij}}{n_{ij} - 6} \right) \left( \frac{n_{ij}}{6} \right)^{6/(n_{ij} - 6)}$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{VDW}(r_{ij}) = - \frac{dE_{VDW}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{VDW}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

Using n-6 LJ potential defined above:

$$F_{VDW}(r_{ij}) = 6C_{n_{ij}} \epsilon_{ij} \left[ \frac{n_{ij}}{6} \times \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \times \frac{1}{r_{ij}}$$

---

**Note:** This option only evaluates the energy up to specified Rcut distance. Tail correction to energy and pressure can be specified to account for infinite cutoff distance.

---

## 12.2 EXP6

This option calculates potential energy without any truncation.

**Potential Calculation** Interactions between atoms can be modeled with an exp-6 (Buckingham) potential,

$$E_{\text{VDW}}(r_{ij}) = \begin{cases} \frac{\alpha_{ij}\epsilon_{ij}}{\alpha_{ij}-6} \left[ \frac{6}{\alpha_{ij}} \exp\left(\alpha_{ij}\left[1 - \frac{r_{ij}}{R_{\min,ij}}\right]\right) - \left(\frac{R_{\min,ij}}{r_{ij}}\right)^6 \right] & r_{ij} \geq R_{\max,ij} \\ \infty & r_{ij} < R_{\max,ij} \end{cases}$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $R_{\min,ij}$  are, respectively, the separation, minimum potential, and minimum potential distance for the pair of interaction sites  $i$  and  $j$ . The constant  $\alpha_{ij}$  is an exponential-6 parameter. The cutoff distance  $R_{\max,ij}$  is the smallest positive value for which  $\frac{dE_{\text{VDW}}(r_{ij})}{dr_{ij}} = 0$ .

---

**Note:** In order to use `Mie` or `Exotice` potential file format for Buckingham potential, instead of defining  $R_{\min}$ , we define  $\sigma$  (collision diameter or the distance, where potential is zero) and GOMC will calculate the  $R_{\min}$  and  $R_{\max}$  using Buckingham potential equation.

---

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{\text{VDW}}(r_{ij}) = -\frac{dE_{\text{VDW}}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{\text{VDW}}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

Using exp-6 potential defined above:

$$F_{\text{VDW}}(r_{ij}) = \begin{cases} \frac{6\alpha_{ij}\epsilon_{ij}}{r_{ij}(\alpha_{ij}-6)} \left[ \frac{r_{ij}}{R_{\min,ij}} \exp\left(\alpha_{ij}\left[1 - \frac{r_{ij}}{R_{\min,ij}}\right]\right) - \left(\frac{R_{\min,ij}}{r_{ij}}\right)^6 \right] & r_{ij} \geq R_{\max,ij} \\ \infty & r_{ij} < R_{\max,ij} \end{cases}$$

---

**Note:** This option only evaluates the energy up to specified `Rcut` distance. Tail correction to energy and pressure can be specified to account for infinite cutoff distance.

---

## 12.3 SHIFT

This option forces the potential energy to be zero at `Rcut` distance.

**Potential Calculation** Interactions between atoms can be modeled with an n-6 potential,

$$E_{\text{VDW}}(r_{ij}) = C_{n_{ij}}\epsilon_{ij} \left[ \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{n_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^6 \right] - C_{n_{ij}}\epsilon_{ij} \left[ \left(\frac{\sigma_{ij}}{r_{\text{cut}}}\right)^{n_{ij}} - \left(\frac{\sigma_{ij}}{r_{\text{cut}}}\right)^6 \right]$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $\sigma_{ij}$  are, respectively, the separation, minimum potential, and collision diameter for the pair of interaction sites  $i$  and  $j$ . The constant  $C_n$  is a normalization factor according to Eq. 3, such that the minimum of the potential remains at  $-\epsilon_{ij}$  for all  $n_{ij}$ . In the 12-6 potential,  $C_n$  reduces to the familiar value of 4.

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{\text{VDW}}(r_{ij}) = -\frac{dE_{\text{VDW}}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{\text{VDW}}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

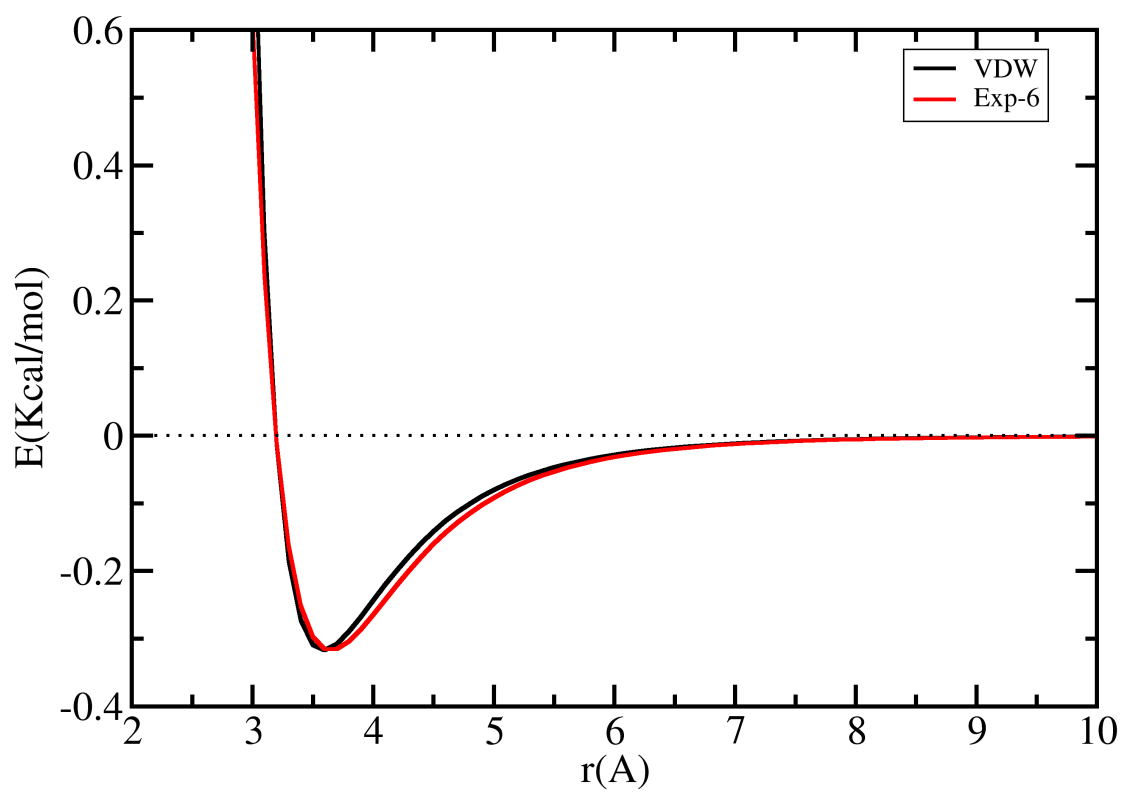


Fig. 1: Graph of Van der Waals interaction for comparison of VDW and EXP6 potentials.

Using SHIFT potential function defined above:

$$F_{VDW}(r_{ij}) = 6C_{n_{ij}}\epsilon_{ij} \left[ \frac{n_{ij}}{6} \times \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \times \frac{1}{r_{ij}}$$

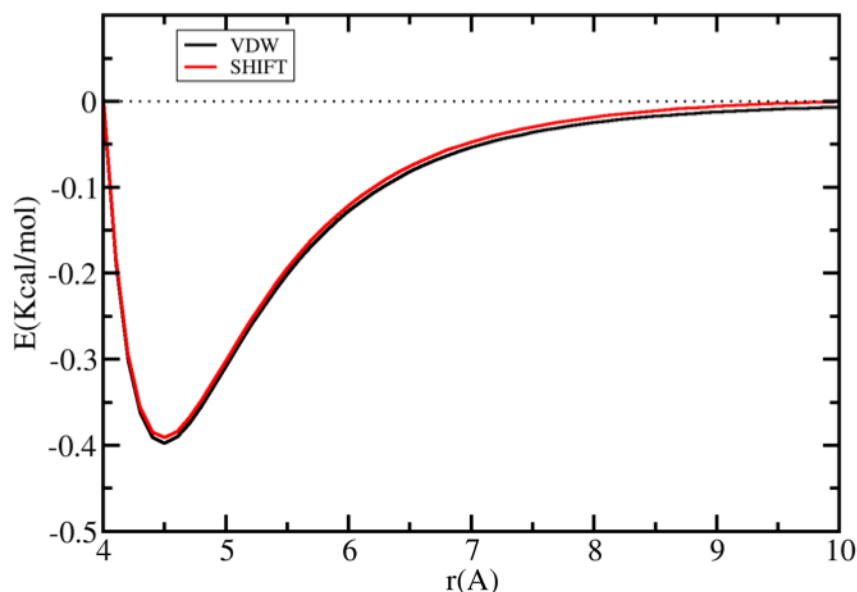


Fig. 2: Graph of Van der Waals potential with and without the application of the SHIFT function. With the SHIFT function active, the potential by force was reduced to 0.0 at the Rcut distance. With the SHIFT function, there is a discontinuity where the potential is truncated.

## 12.4 SWITCH

This option in CHARMM or EXOTIC force field smoothly forces the potential energy to be zero at Rcut distance and starts modifying the potential at Rswitch distance.

**Potential Calculation** Interactions between atoms can be modeled with an n-6 potential,

$$E_{VDW}(r_{ij}) = C_{n_{ij}}\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \times \varphi_E(r_{ij})$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $\sigma_{ij}$  are, respectively, the separation, minimum potential, and collision diameter for the pair of interaction sites  $i$  and  $j$ . The constant  $C_n$  is a normalization factor according to Eq. 3, such that the minimum of the potential remains at  $-\epsilon_{ij}$  for all  $n_{ij}$ . In the 12-6 potential,  $C_n$  reduces to the familiar value of 4.

The factor  $\varphi_E$  is defined as:

$$\varphi_E(r_{ij}) = \begin{cases} 1 & r_{ij} \leq r_{switch} \\ \frac{(r_{cut}^2 - r_{ij}^2)^2 \times (r_{cut}^2 - 3r_{switch}^2 + 2r_{ij}^2)}{(r_{cut}^2 - r_{switch}^2)^3} & r_{switch} < r_{ij} < r_{cut} \\ 0 & r_{ij} \geq r_{cut} \end{cases}$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{VDW}(r_{ij}) = -\frac{dE_{VDW}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{VDW}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

Using SWITCH potential function defined above:

$$F_{VDW}(r_{ij}) = \left[ 6C_{n_{ij}}\epsilon_{ij} \left[ \frac{n_{ij}}{6} \times \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \times \frac{\varphi_E(r_{ij})}{r_{ij}} - C_{n_{ij}}\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{n_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \times \varphi_F(r_{ij}) \right]$$

The factor  $\varphi_F$  is defined as:

$$\varphi_F(r_{ij}) = \begin{cases} 0 & r_{ij} \leq r_{switch} \\ \frac{12r_{ij}(r_{cut}^2 - r_{ij}^2) \times (r_{switch}^2 - r_{ij}^2)}{(r_{cut}^2 - r_{switch}^2)^3} & r_{switch} < r_{ij} < r_{cut} \\ 0 & r_{ij} \geq r_{cut} \end{cases}$$

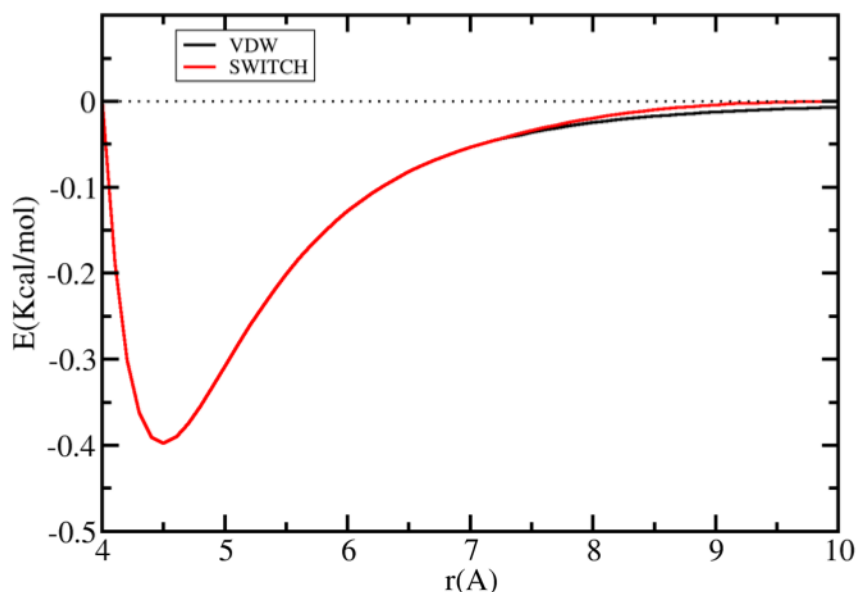


Fig. 3: Graph of Van der Waals potential with and without the application of the SWITCH function. With the SWITCH function active, the potential is smoothly reduced to 0.0 at the Rcut distance.

## 12.5 SWITCH (MARTINI)

This option in MARTINI force field smoothly forces the potential energy to be zero at Rcut distance and starts modifying the potential at Rswitch distance.

**Potential Calculation** Potential Calculation: Interactions between atoms can be modeled with an n-6 potential. In standard MARTINI,  $n$  is equal to 12,

$$E_{VDW}(r_{ij}) = C_{n_{ij}}\epsilon_{ij} \left[ \sigma_{ij}^n \left( \frac{1}{r_{ij}^n} + \varphi_{E,n}(r_{ij}) \right) - \sigma_{ij}^6 \left( \frac{1}{r_{ij}^6} + \varphi_{E,6}(r_{ij}) \right) \right]$$

where  $r_{ij}$ ,  $\epsilon_{ij}$ , and  $\sigma_{ij}$  are, respectively, the separation, minimum potential, and collision diameter for the pair of interaction sites  $i$  and  $j$ . The constant  $C_n$  is a normalization factor according to Eq. 3, such that the minimum of the potential remains at  $-\epsilon_{ij}$  for all  $n_{ij}$ . In the 12-6 potential,  $C_n$  reduces to the familiar value of 4.

The factor  $\varphi_{E,\alpha}$  and constants are defined as:

$$\varphi_{E,\alpha}(r_{ij}) = \begin{cases} -C_\alpha & r_{ij} \leq r_{switch} \\ -\frac{A_\alpha}{3}(r_{ij} - r_{switch})^3 - \frac{B_\alpha}{4}(r_{ij} - r_{switch})^4 - C_\alpha & r_{switch} < r_{ij} < r_{cut} \\ 0 & r_{ij} \geq r_{cut} \end{cases}$$

$$A_\alpha = \alpha \frac{(\alpha + 1)r_{switch} - (\alpha + 4)r_{cut}}{r_{cut}^{(\alpha+2)}(r_{cut} - r_{switch})^2}$$

$$B_\alpha = \alpha \frac{(\alpha + 1)r_{switch} - (\alpha + 3)r_{cut}}{r_{cut}^{(\alpha+2)}(r_{cut} - r_{switch})^3}$$

$$C_\alpha = \frac{1}{r_{cut}^\alpha} - \frac{A_\alpha}{3}(r_{cut} - r_{switch})^3 - \frac{B_\alpha}{4}(r_{cut} - r_{switch})^4$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{VDW}(r_{ij}) = -\frac{dE_{VDW}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{VDW}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

Using the SWITCH potential function defined for MARTINI force field:

$$F_{VDW}(r_{ij}) = C_{n_{ij}}\epsilon_{ij} \left[ \sigma_{ij}^n \left( \frac{n}{r_{ij}^{(n+1)}} + \varphi_{F,n}(r_{ij}) \right) - \sigma_{ij}^6 \left( \frac{6}{r_{ij}^{(6+1)}} + \varphi_{F,6}(r_{ij}) \right) \right]$$

The constants defined in Eq. 14-16 and the factor  $\varphi_{F,\alpha}$  defined as:

$$\varphi_{F,\alpha}(r_{ij}) = \begin{cases} 0 & r_{ij} \leq r_{switch} \\ A_\alpha(r_{ij} - r_{switch})^2 + B_\alpha(r_{ij} - r_{switch})^3 & r_{switch} < r_{ij} < r_{cut} \\ 0 & r_{ij} \geq r_{cut} \end{cases}$$

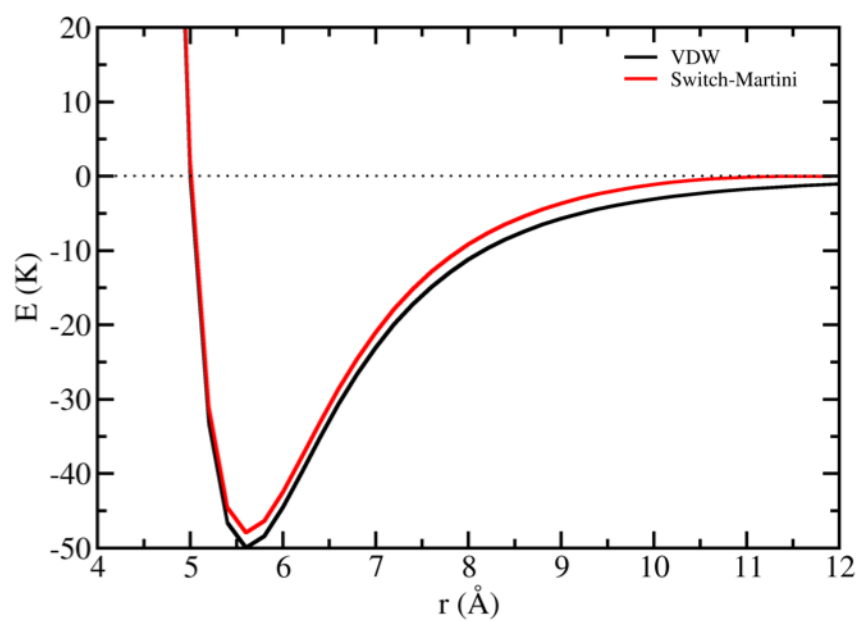


Fig. 4: Graph of Van der Waals potential with and without the application of the SWITCH function in MARTINI force field. With the SWITCH function active, the potential is smoothly reduced to 0.0 at the Rcut distance.





## INTERMOLECULAR ENERGY AND VIRIAL FUNCTION (ELECTROSTATIC)

In this section, the virial and energy equation of electrostatic interaction for different potential function are discussed in details.

### 13.1 Ewald

This option calculate electrostatic energy using standard *Ewald Summation Method*.

---

**Note:** Once this option is activated, it would override the the electrostatic calculation using VDW, EXP6, SHIFT, and SWITCH functions.

---

**Potential Calculation** Coulomb interactions between atoms can be modeled as

$$E(\text{Ewald}) = E_{real} + E_{reciprocal} + E_{self} + E_{correction}$$

$E_{real}$ : Defines the short range electrostatic energy according to

$$E_{real} = \frac{1}{4\pi\epsilon_0} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N q_i q_j \frac{\text{erfc}(\alpha r_{ij})}{r_{ij}}$$

, where  $\alpha$  is Ewald separation parameter according to

$$\alpha = \frac{\sqrt{-\log(\text{Tolerance})}}{r_{cut}}$$

, where Tolerance is a parameter, controlling the desired accuracy.

$E_{reciprocal}$ : Defines the long range electrostatic energy according to,

$$E_{reciprocal} = \frac{1}{\epsilon_0 V} \frac{1}{2} \sum_{\vec{k} \neq 0} \frac{1}{k^2} \exp\left(\frac{-\vec{k}^2}{4\alpha^2}\right) \left[ |R_{sum}|^2 + |I_{sum}|^2 \right]$$

, where  $\vec{k}$  is reciprocal vector,  $R_{sum}$  and  $I_{sum}$  are,

$$R_{sum} = \sum_{i=1}^N q_i \cos(\vec{k} \cdot \vec{x}_i)$$

$$I_{sum} = \sum_{i=1}^N q_i \sin(\vec{k} \cdot \vec{x}_i)$$

$E_{self}$ : Defines the self energy according to,

$$E_{self} = -\frac{\alpha}{4\pi\epsilon_0\sqrt{\pi}} \sum_{i=1}^N q_i^2$$

$E_{correction}$ : Defines intra-molecule nonbonded energy,

$$E_{correction} = -\frac{1}{4\pi\epsilon_0} \frac{1}{2} \sum_{j=1}^N \sum_{l=1}^{N_j} \sum_{m=1}^{N_j} q_{jl} q_{jm} \frac{erf(\alpha r_{jlm})}{r_{jlm}}$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance, Eq. 4. Coulomb force between atoms can be modeled as,

$$W_{Ewald} = W_{real} + W_{reciprocal}$$

$W_{real}$  defines the short range electrostatic and  $W_{reciprocal}$  defines the long range electrostatic force according to,

$$W_{real} = \frac{1}{4\pi\epsilon_0} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N q_i q_j \left[ \frac{erfc(\alpha r_{ij})}{r_{ij}} + \frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2 r_{ij}^2) \right] \times \frac{\vec{r}_{ij}}{r_{ij}^2}$$

$$W_{reciprocal} = \frac{1}{\epsilon_0 V} \frac{1}{2} \sum_{\vec{k} \neq 0} \left[ \frac{1}{\vec{k}^2} \exp\left(\frac{-\vec{k}^2}{4\alpha^2}\right) \left( |R_{sum}|^2 + |I_{sum}|^2 \right) \left( 1 - \frac{\vec{k}^2}{2\alpha^2} \right) \right] +$$

$$\sum_{i=1}^N \frac{1}{\epsilon_0 V} \sum_{\vec{k} \neq 0} \left[ \frac{q_i}{\vec{k}^2} \exp\left(\frac{-\vec{k}^2}{4\alpha^2}\right) \left[ I_{sum} \times \cos(\vec{k} \cdot \vec{x}_i) - R_{sum} \times \sin(\vec{k} \cdot \vec{x}_i) \right] \right] \times (\vec{k} \cdot \vec{r}_{ic})$$

, where  $\vec{r}_{ic}$  is the vector between atom and the center of the mass of the molecule.

## 13.2 VDW

Using VDW potential type without Ewald method, simply uses coulomb energy to calculate the electrostatic potential.

**Potential Calculation** Coulomb interactions between atoms can be modeled as

$$E_{Elect}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{Elect}(r_{ij}) = -\frac{dE_{Elect}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{Elect}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

$$F_{Elect}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} \left( \frac{1}{r_{ij}^2} \right)$$

## 13.3 EXP6

Using EXP6 potential type without Ewald method, simply uses coulomb energy to calculate the electrostatic potential.

**Potential Calculation** Coulomb interactions between atoms can be modeled as

$$E_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{\text{Elect}}(r_{ij}) = -\frac{dE_{\text{Elect}}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{\text{Elect}}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

$$F_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} \left( \frac{1}{r_{ij}^2} \right)$$

## 13.4 SHIFT

This option forces the electrostatic energy to be zero at Rcut distance.

**Potential Calculation** Coulomb interactions between atoms can be modeled as

$$E_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} \left( \frac{1}{r_{ij}} - \frac{1}{r_{\text{cut}}} \right)$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{\text{Elect}}(r_{ij}) = -\frac{dE_{\text{Elect}}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{\text{Elect}}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

$$F_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} \left( \frac{1}{r_{ij}^2} \right)$$

## 13.5 SWITCH

This option in CHARMM or EXOTIC force field forces the electrostatic energy to be zero at Rcut distance.

**Potential Calculation** Coulomb interactions between atoms can be modeled as,

$$E_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} \left( \left( \frac{r_{ij}}{r_{\text{cut}}} \right)^2 - 1.0 \right)^2 \frac{1}{r_{ij}}$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{\text{Elect}}(r_{ij}) = -\frac{dE_{\text{Elect}}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{\text{Elect}}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

$$F_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} \left[ \left( \left( \frac{r_{ij}}{r_{\text{cut}}} \right)^2 - 1.0 \right)^2 \frac{1}{r_{ij}^2} - \left( \frac{4}{r_{\text{cut}}^2} \right) \left( \left( \frac{r_{ij}}{r_{\text{cut}}} \right)^2 - 1.0 \right) \right]$$

## 13.6 SWITCH (MARTINI)

This option in MARTINI force field smoothly forces the potential energy to be zero at Rcut distance and starts modifying the potential at Rswitch = 0.0 distance.

**Potential Calculation** Coulomb interactions between atoms can be modeled as,

$$E_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_1} \left( \frac{1}{r_{ij}} + \varphi_{E,1}(r_{ij}) \right)$$

, where  $\epsilon_1$  is the dielectric constant, which in MARTINI force field is equal to 15.0 and  $\varphi_{E,\alpha=1}(r_{ij})$  is defined as:

$$\varphi_{E,\alpha}(r_{ij}) = \begin{cases} -C_\alpha & r_{ij} \leq r_{\text{switch}} \\ -\frac{A_\alpha}{3}(r_{ij} - r_{\text{switch}})^3 - \frac{B_\alpha}{4}(r_{ij} - r_{\text{switch}})^4 - C_\alpha & r_{\text{switch}} < r_{ij} < r_{\text{cut}} \\ 0 & r_{ij} \geq r_{\text{cut}} \end{cases}$$

$$A_\alpha = \alpha \frac{(\alpha + 1)r_{\text{switch}} - (\alpha + 4)r_{\text{cut}}}{r_{\text{cut}}^{(\alpha+2)}(r_{\text{cut}} - r_{\text{switch}})^2}$$

$$B_\alpha = \alpha \frac{(\alpha + 1)r_{\text{switch}} - (\alpha + 3)r_{\text{cut}}}{r_{\text{cut}}^{(\alpha+2)}(r_{\text{cut}} - r_{\text{switch}})^3}$$

$$C_\alpha = \frac{1}{r_{\text{cut}}^\alpha} - \frac{A_\alpha}{3}(r_{\text{cut}} - r_{\text{switch}})^3 - \frac{B_\alpha}{4}(r_{\text{cut}} - r_{\text{switch}})^4$$

**Virial Calculation** Virial is basically the negative derivative of energy with respect to distance, multiplied by distance.

$$W_{\text{Elect}}(r_{ij}) = -\frac{dE_{\text{Elect}}(r_{ij})}{dr_{ij}} \times \frac{\vec{r}_{ij}}{r_{ij}} = F_{\text{Elect}}(r_{ij}) \times \frac{\vec{r}_{ij}}{r_{ij}}$$

$$F_{\text{Elect}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_1} \left( \frac{1}{r_{ij}^2} + \varphi_{F,1}(r_{ij}) \right)$$

, where  $\varphi_{F,\alpha=1}(r_{ij})$  is defined as:

$$\varphi_{F,\alpha}(r_{ij}) = \begin{cases} 0 & r_{ij} \leq r_{\text{switch}} \\ A_\alpha(r_{ij} - r_{\text{switch}})^2 + B_\alpha(r_{ij} - r_{\text{switch}})^3 & r_{\text{switch}} < r_{ij} < r_{\text{cut}} \\ 0 & r_{ij} \geq r_{\text{cut}} \end{cases}$$

## LONG-RANGE CORRECTION (ENERGY AND VIRIAL)

To accelerate the simulation performance, the nonbonded potential is usually truncated at specific cut-off (Rcut) distance. To compensate the missing potential energy and force, beyond the Rcut distance, the long-range correction (LRC) or tail correction to energy and virial must be calculated and added to total energy and virial of the system, to account for infinite cutoff distance.

The VDW and EXP6 energy functions, evaluates the energy up to specified Rcut distance. In this section, the LRC equations for virial and energy term for Van der Waals interaction are discussed in details.

### 14.1 VDW

This option calculates potential energy using standard Lennard Jones (12-6) or Mie (n-6) potentials, up to specific Rcut distance.

**Energy** For homogeneous system, the long-range correction energy can be analytically calculated:

$$E_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} \int_{r=r_{\text{cut}}}^{\infty} r^2 E_{\text{VDW}}(r) dr$$

$$E_{\text{VDW}}(r) = C_n \epsilon \left[ \left( \frac{\sigma}{r} \right)^n - \left( \frac{\sigma}{r} \right)^6 \right]$$

where  $N$ ,  $V$ ,  $r$ ,  $\epsilon$ , and  $\sigma$  are the number of molecule, volume of the system, separation, minimum potential, and collision diameter, respectively. The constant  $C_n$  is a normalization factor such that the minimum of the potential remains at  $-\epsilon$  for all  $n$ . In the 12-6 potential,  $C_n$  reduces to the familiar value of 4.

$$C_n = \left( \frac{n}{n-6} \right) \left( \frac{n}{6} \right)^{6/(n-6)}$$

Substituting the general Lennard Jones energy equation into the integral, the long-range correction energy term is defined by:

$$E_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} C_n \epsilon \sigma^3 \left[ \frac{1}{n-3} \left( \frac{\sigma}{r_{\text{cut}}} \right)^{(n-3)} - \frac{1}{3} \left( \frac{\sigma}{r_{\text{cut}}} \right)^3 \right]$$

**Virial** For homogeneous system, the long-range correction virial can be analytically calculated:

$$W_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} \int_{r=r_{\text{cut}}}^{\infty} r^3 F_{\text{VDW}}(r) dr$$

$$F_{\text{VDW}}(r) = \frac{6C_n \epsilon}{r} \left[ \frac{n}{6} \times \left( \frac{\sigma}{r} \right)^n - \left( \frac{\sigma}{r} \right)^6 \right]$$

Substituting the general Lennard Jones force equation into the integral, the long-range correction virial term is defined by:

$$W_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} C_n \epsilon \sigma^3 \left[ \frac{n}{n-3} \left( \frac{\sigma}{r_{\text{cut}}} \right)^{(n-3)} - 2 \left( \frac{\sigma}{r_{\text{cut}}} \right)^3 \right]$$

## 14.2 EXP6

This option calculates potential energy using Buckingham potentials, up to specific Rcut distance.

**Energy** For homogeneous system, the long-range correction energy can be analytically calculated:

$$E_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} \int_{r=r_{\text{cut}}}^{\infty} r^2 E_{\text{VDW}}(r) dr$$

$$E_{\text{VDW}}(r) = \begin{cases} \frac{\alpha \epsilon}{\alpha - 6} \left[ \frac{6}{\alpha} \exp \left( \alpha \left[ 1 - \frac{r}{R_{\text{min}}} \right] \right) - \left( \frac{R_{\text{min}}}{r} \right)^6 \right] & r \geq R_{\text{max}} \\ \infty & r < R_{\text{max}} \end{cases}$$

where  $r$ ,  $\epsilon$ , and  $R_{\text{min}}$  are, respectively, the separation, minimum potential, and minimum potential distance. The constant  $\alpha$  is an exponential-6 parameter. The cutoff distance  $R_{\text{max}}$  is the smallest positive value for which  $\frac{dE_{\text{VDW}}(r)}{dr} = 0$ .

Substituting the Buckingham potential into the integral, the long-range correction energy term is defined by:

$$E_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} \left[ AB \exp \left( \frac{-r_{\text{cut}}}{B} \right) \left( 2B^2 + 2Br_{\text{cut}} + r_{\text{cut}}^2 \right) - \frac{C}{3r_{\text{cut}}^3} \right]$$

$$A = \frac{6\epsilon \exp(\alpha)}{\alpha - 6}$$

$$B = \frac{R_{\text{min}}}{\alpha}$$

$$C = \frac{\epsilon \alpha R_{\text{min}}^6}{\alpha - 6}$$

**Virial** For homogeneous system, the long-range correction virial can be analytically calculated:

$$W_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} \int_{r=r_{\text{cut}}}^{\infty} r^3 F_{\text{VDW}}(r) dr$$

$$F_{\text{VDW}}(r) = \begin{cases} \frac{6\alpha\epsilon}{r(\alpha-6)} \left[ \frac{r}{R_{\text{min}}} \exp \left( \alpha \left[ 1 - \frac{r}{R_{\text{min}}} \right] \right) - \left( \frac{R_{\text{min}}}{r} \right)^6 \right] & r \geq R_{\text{max}} \\ \infty & r < R_{\text{max}} \end{cases}$$

Substituting the Buckingham potential into the integral, the long-range correction virial term is defined by:

$$W_{\text{LRC(VDW)}} = \frac{2\pi N^2}{V} \left[ A \exp \left( \frac{-r_{\text{cut}}}{B} \right) \left( 6B^3 + 6B^2 r_{\text{cut}} + 3Br_{\text{cut}}^2 + r_{\text{cut}}^3 \right) - \frac{2C}{3r_{\text{cut}}^3} \right]$$

## COUPLING INTERACTION WITH $\lambda$

In this section, the scaling nonbonded and long-range correction energies with  $\lambda$  is discussed in detailed.

$$E_{\lambda} = E_{\lambda}(\text{VDW}) + E_{\lambda}(\text{Elect}) + E_{\lambda}(\text{LRC-VDW}) + E_{\lambda}(\text{LRC-Elect})$$

### 15.1 VDW

#### 15.1.1 Soft-core

In free energy calculation, the VDW interaction between solute and solvent is scaled with  $\lambda$ , non-linearly (soft-core scheme), to avoid end-point catastrophe and numerical issue

$$E_{\lambda}(\text{VDW}) = \lambda_{\text{VDW}} E_{\text{VDW}}(r_{sc})$$

the scaled solute-solvent distance,  $r_{sc}$  is defined as:

$$r_{sc} = \left[ \alpha (1 - \lambda_{\text{VDW}})^p \sigma^6 + r^6 \right]^{\frac{1}{6}}$$

where,  $\alpha$  and  $p$  are the soft-core parameters defined by user (ScaleAlpha, ScalePower) and  $\sigma$  is the diameter of atom. To improve numerical convergence of the calculation, a minimum interaction diameter  $\sigma_{min}$  should be defined by user (MinSigma) for any atom with a diameter less than  $\sigma_{min}$ , e.g. hydrogen atoms attached to oxygen in water or alcohols.

To calculate the solvation free energy with thermodynamic integration (TI) method, the derivative of energy with respect to lambda ( $\frac{dE_{\lambda}(\text{VDW})}{d\lambda_{\text{VDW}}}$ ) is required:

$$\frac{dE_{\lambda}(\text{VDW})}{d\lambda_{\text{VDW}}} = E_{\text{VDW}}(r_{sc}) + \frac{p\alpha\lambda_{\text{VDW}}}{6} \left( 1 - \lambda_{\text{VDW}} \right)^{p-1} \left( \frac{\sigma^6}{r_{sc}^5} \right) F_{\text{VDW}}(r_{sc})$$

### 15.2 Electrostatic

#### 15.2.1 Hard-core

In free energy calculation, the Coulombic interaction between solute and solvent can be scaled with  $\lambda$ , **linearly** (hard-core scheme), by setting the ScaleCoulomb to false.

$$E_{\lambda}(\text{Elect}) = \lambda_{\text{Elect}} E_{\text{Elect}}(r)$$

where,  $r$  is the distance between solute and solvent, without any modification.

To calculate the solvation free energy with thermodynamic integration (TI) method, the derivative of energy with respect to lambda ( $\frac{dE_{\lambda}(\text{Elect})}{d\lambda_{\text{Elect}}}$ ) is required:

$$\frac{dE_{\lambda}(\text{Elect})}{d\lambda_{\text{Elect}}} = E_{\text{Elect}}(r)$$

**Warning:** To avoid end-point catastrophe and numerical issue, it's suggested to turn on the VDW interaction completely, before turning on the Coulombic interaction.

## 15.2.2 Soft-core

In free energy calculation, the Coulombic interaction between solute and solvent can be scaled with  $\lambda$ , **non-linearly** (soft-core scheme), to avoid end-point catastrophe and numerical issue. This option can be activated by setting the `ScaleCoulomb` to true.

$$E_{\lambda}(\text{Elect}) = \lambda_{\text{Elect}} E_{\text{Elect}}(r_{sc})$$

the scaled solute-solvent distance,  $r_{sc}$  is defined as:

$$r_{sc} = \left[ \alpha (1 - \lambda_{\text{Elect}})^p \sigma^6 + r^6 \right]^{\frac{1}{6}}$$

where,  $\alpha$  and  $p$  are the soft-core parameters defined by user (`ScaleAlpha`, `ScalePower`) and  $\sigma$  is the diameter of atom. To improve numerical convergence of the calculation, a minimum interaction diameter  $\sigma_{min}$  should be defined by user (`MinSigma`) for any atom with a diameter less than  $\sigma_{min}$ , e.g. hydrogen atoms attached to oxygen in water or alcohols.

To calculate the solvation free energy with thermodynamic integration (TI) method, the derivative of energy with respect to lambda ( $\frac{dE_{\lambda}(\text{Elect})}{d\lambda_{\text{Elect}}}$ ) is required:

$$\frac{dE_{\lambda}(\text{Elect})}{d\lambda_{\text{Elect}}} = E_{\text{Elect}}(r_{sc}) + \frac{p\alpha\lambda_{\text{Elect}}}{6} \left(1 - \lambda_{\text{Elect}}\right)^{p-1} \left(\frac{\sigma^6}{r_{sc}^5}\right) E_{\text{Elect}}(r_{sc})$$

**Warning:** Using soft-core scheme to scale the coulombic interaction non-linearly, would result in **inaccurate** results if `Ewald` method is activated.

Using *Ewald Summation Method*, we suggest to use hard-core scheme, to scale the coulombic interaction linearly with  $\lambda$ .

## 15.3 Long-range Correction (VDW)

The effect of long-range corrections on predicted free energies were determined for VDW interactions via a linear coupling with  $\lambda$ .

$$E_{\lambda}(\text{LRC-VDW}) = \lambda_{\text{VDW}} \Delta E_{\text{LRC}(\text{VDW})}$$

where,  $\Delta E_{\text{LRC}(\text{VDW})}$  is the the change in the long-range correction energy, due to adding a fully interacting solute to the solvent for VDW interaction.

To calculate the solvation free energy with thermodynamic integration (TI) method, the derivative of energy with respect to lambda ( $\frac{dE_{\lambda}(\text{LRC-VDW})}{d\lambda_{\text{VDW}}}$ ) is required:

$$\frac{dE_{\lambda}(\text{LRC-VDW})}{d\lambda_{\text{VDW}}} = \Delta E_{\text{LRC}(\text{VDW})}$$



## 15.4 Long-range Correction (Electrostatic)

Using *Ewald Summation Method*, the effect of long-range corrections on predicted free energies were determined for Coulombic interactions via a linear coupling with  $\lambda$ .

$$E_{\lambda}(\text{LRC-Elect}) = \lambda_{\text{Elect}} \left[ \Delta E_{\text{reciprocal}} + \Delta E_{\text{self}} + \Delta E_{\text{correction}} \right]$$

where,  $\Delta E_{\text{reciprocal}}$ ,  $\Delta E_{\text{self}}$ , and  $\Delta E_{\text{correction}}$  are the the change in the reciprocal, self, and correction energy term in Ewald method, due to adding a fully interacting solute to the solvent.

To calculate the solvation free energy with thermodynamic integration (TI) method, the derivative of energy with respect to lambda ( $\frac{dE_{\lambda}(\text{LRC-Elect})}{d\lambda_{\text{Elect}}}$ ) is required:

$$\frac{dE_{\lambda}(\text{LRC-Elect})}{d\lambda_{\text{Elect}}} = \Delta E_{\text{reciprocal}} + \Delta E_{\text{self}} + \Delta E_{\text{correction}}$$



## HYBRID MONTE CARLO-MOLECULAR DYNAMICS (MCMD)

In this section, the tips and tricks to get a hybrid MCMD simulation with GOMC and NAMD running are discussed. Most of these issues will be handled by the scripts provided with py-MCMD, but the concerns are raised here for users interested in setting up custom systems. Careful attention should be made to ensure the system is centered in the first octant of 3D space, originates at  $[\text{boxlength}/2, \text{boxlength}/2, \text{boxlength}/2]$ , and the box length exceeds the radius of gyration of all molecules.

Link to documentation: <https://py-mcmd.readthedocs.io/en/latest/>

Link to Github Repository: <https://github.com/GOMC-WSU/py-MCMD>

### 16.1 GOMC Requirements

GOMC currently requires that Box length / 2 exceed the radius of gyration of all molecules in the system.

Grand-Canonical Molecular Dynamics (GCMD) or Gibbs Ensemble with Molecular Dynamics changes the number of molecules in each box. This will alter the ordering of the molecules, posing a challenge when the user tries to concatenate the trajectories or follow one atom through a trajectory.

The GOMC checkpoint file will reload the molecules in the original order to ensure the GOMC trajectories (PDB/DCD) have a consistent ordering for analysis. Atoms that are not currently in a specific box are given the coordinate (0.00, 0.00, 0.00). The occupancy value corresponds to the box a molecule is currently in (e.g. 0.00 for box 0; 1.00 for box 1).

The restart binary coordinates, velocities, and box dimensions (xsc) from NAMD need to be loaded along with the checkpoint file, restart PDB, and restart PSF from the previous GOMC cycle.

The python script from the py-MCMD git repository, `combine_data_NAMD_GOMC.py`, requires the GOMC step reset to 0 every cycle

InitStep	0
----------	---

### 16.2 NAMD Requirements

GOMC outputs all the files needed to continue a simulation box in Molecular Dynamics (pdb, psf, xsc, coor, vel, xsc). These files should all be used.

There are certain flexibilities that NAMD allows for that GOMC doesn't support. To ensure the two systems are compatible the following settings in the NAMD configuration file are required:

Rigid bonds, since GOMC doesn't support bond length sampling.

rigidBonds	all
------------	-----

Fixed volume, since GOMC maintains the origin of the box at [box length/2, box length/2, box length/2]

```
# Constant Pressure Control (variable volume)
```

```
langevinPiston      off
```

```
useGroupPressure    yes
```

```
useFlexibleCell     no
```

```
useConstantArea     no
```

Box origin must be centered at [box length/2, box length/2, box length/2]

cell0origin	x_box length/2	y_box length/2	z_box length/2
-------------	----------------	----------------	----------------

## 16.3 Dynamic Subvolumes for Dual Control Volume Molecular Dynamics

To define a subvolume in the simulation, use the subvolume keywords to choose an subvolume id, center, either the geometric center of a list of atoms or absolute cartesian coordinate, and dimensions. The residues that can be inserted/deleted in the subvolume, custom chemical potential, and periodicity of the subvolume may also be specified. Fugacity can be replaced for chemical potential. A chemical gradient can be established in the simulation by defining two or more subvolume with different chemical potentials of a given residue. After the molecule is inserted/deleted within one subvolume, it diffuses out and is inserted/deleted from the other at a different chemical potential, forming a gradient.

To define two control volumes forming a gradient from the left to the right of the box

SubVolumeBox	0	0
SubVolumeDim	0	left_one_fifth y_dim_box_0 z_dim_box_0
SubVolumeResidueKind	0	DIOX
SubVolumeRigidSwap	0	true
SubVolumeCenter	0	left_center y_origin_box z_origin_box
SubVolumePBC	0	XYZ
SubVolumeChemPot	0	DIOX -2000
SubVolumeBox	1	0
SubVolumeDim	1	right_one_fifth y_dim_box_0 z_dim_box_0
SubVolumeResidueKind	1	DIOX

(continues on next page)

(continued from previous page)

SubVolumeRigidSwap	1	true	
SubVolumeCenter	1	right_center	y_origin_box z_origin_box
SubVolumePBC	1	XYZ	
SubVolumeChemPot	1	DIOX	-4000

## 16.4 Run a Hybrid Monte Carlo-Molecular Dynamics Sim

GOMC and NAMD produce compatible input/output files, which allow the system to alternate between Monte Carlo and Molecular Dynamics. The py-MCMD script automates the directory generation, running of GOMC and NAMD, and concatenation of the short alternating runs. Simulating the Grand Canonical ensemble in GOMC with only molecule transfers, allows the MD simulations to continue where they left off, with a varying number of molecules.

Refer to the section on Hybrid Monte Carlo-Molecular Dynamics in the manual and attached links.

Link to documentation: <https://py-mcmd.readthedocs.io/en/latest/>

Link to Github Repository: <https://github.com/GOMC-WSU/py-MCMD>

```
$ git clone https://github.com/GOMC-WSU/py-MCMD.git
$ cd py-MCMD
$ python run_NAMD_GOMC.py -f user_input_NAMD_GOMC.json
$ python combine_data_NAMD_GOMC.py -f user_input_combine_data_NAMD_GOMC.json
```



## HOW TO?

In this section, we are providing a summary of what actions or modification need to be done in order to answer your simulation problem.

### 17.1 Visualizing Simulation

If `CoordinatesFreq` is enabled in configuration file, GOMC will output the molecule coordinates every specified stpes. The PDB and PSF output (merging of atom entries) has already been mentioned/explained in previous sections. To recap: The PDB file's ATOM entries' occupancy is used to represent the box the molecule is in for the current frame. All molecules are listed in order in which they were read (i.e. if box 0 has 1, 2, ...,  $N1$  molecules and box 1 has 1, 2, ...,  $N2$  molecules, then all of the molecules in box 0 are listed first and all the molecules in box 1, i.e. 1, 2, ...,  $N1$ ,  $N1 + 1$ , ...,  $N1 + N2$ ). PDB frames are written as standard PDBs to consecutive file frames.

To visualize, open the output PDB and PSF files by GOMC using VMD, type this command in the terminal:

For all simulation except Gibbs ensemble that has one simulation box:

```
$ vmd ISB_T_270_k_merged.psf ISB_T_270_k_BOX_0.pdb
```

For Gibbs ensemble, visualizing the first box:

```
$ vmd ISB_T_270_k_merged.psf ISB_T_270_k_BOX_0.pdb
```

For Gibbs ensemble, visualizing the second box:

```
$ vmd ISB_T_270_k_merged.psf ISB_T_270_k_BOX_1.pdb
```

---

**Note:** Restart coordinate file (`OutputName_BOX_0_restart.pdb`) cannot be visualize using merged psf file, because atom number does not match. However, you can still open it in vmd using following command and vmd will automatically find the bonds of the molecule based on the coordinates.

---

```
$ vmd ISB_T_270_k_BOX_0_restart.pdb
```

## 17.2 Build molecule and topology file

There are many open-source software that can build a molecule for you, such as [Avagadro](#) , [molefacture](#) in VMD and more. Here we use molefacture features to not only build a molecule, but also creating the topology file.

### 17.2.1 Regular molecule

First, make sure that VMD is installed on your computer. Then, to learn how to build a single PDB file and topology file for united atom butane molecule, please refer to this [document](#) .

We encourage to try to go through our workshop materials:

- To try two days workshop, execute the following command in your terminal to clone the workshop:

```
$ git clone https://github.com/GOMC-WSU/Workshop.git --branch master --single-branch
$ cd Workshop
```

or simply download it from [GitHub](#) .

- To try two hours workshop, execute the following command in your terminal to clone the workshop:

```
$ git clone https://github.com/GOMC-WSU/Workshop.git --branch AICHe --single-branch
$ cd Workshop
```

or simply download it from [GitHub](#) .

### 17.2.2 Molecule with dummy atoms

To simulate a molecule that includes one or more atoms with electrostatic interaction only and no LJ interaction (i.e. dummy atom near of the oxygen along the bisector of the HOH angle in [TIP4P water model](#)), we must perform the following steps to define the dummy atom/atoms:

1. Create a PDB file for single water molecule atoms (H1, O, H2) and a dummy atom (M, in this example), where dummy atom located at 0.150 Å of oxygen and along the bisector of the H1-O-H2 angle.

```
CRYST1    0.000    0.000    0.000  90.00  90.00  90.00 P 1          1
ATOM      1  O  TIP4    1      -0.189    1.073    0.000    0.00    0.00          O
ATOM      2  H1 TIP4    1       0.768    1.114    0.000    0.00    0.00          H
ATOM      3  H2 TIP4    1      -0.469    1.988    0.000    0.00    0.00          H
ATOM      4  M  TIP4    1      -0.102    1.195    0.000    0.00    0.00          D
END
```

2. Pack your desire number of TIP4 water molecule in a box using packmol, as explained before.
3. Include the dummy atom (M) and its charge in your topology file. Define a bond between oxygen and dummy atom. Use vmd and build script to generate your PSF files.

```
* Custom top file -- TIP4P water

MASS    1  OH    15.9994  O !
MASS    2  HO     1.0080  H !
MASS    3  MO     0.0000  D ! Dummy atom for TIP4P model
```

(continues on next page)



(continued from previous page)

```

DEFA FIRS none LAST none
AUTOGENERATE ANGLES DIHEDRALS

RESI TIP4          0.0000 ! TIP4P water
GROUP
ATOM O      OH      0.0000 !      O
ATOM H1     HO      0.5564 !      /  |  \
ATOM H2     HO      0.5564 !      /  M  \
ATOM M      MO     -1.1128 !  H1      H2
BOND  O H1  O H2  O M
PATCHING FIRS NONE LAST NONE

END

```

4. Define all bonded parameters (bond, angles, and dihedral) and nonbonded parameters in your parameter file.

```

*parameteres for TIP4P

BONDS
!
!V(bond) = Kb(b - b0)**2
!
!atom type      Kb      b0
OH  HO  9999999999  0.9572 ! TIP4P O-H bond length
OH  MO  9999999999  0.1500 ! TIP4P M-O bond length

ANGLES
!
!V(angle) = Ktheta(Theta - Theta0)**2
!
!atom types      Ktheta      Theta0
HO  OH  HO  999999999999  104.52 ! H-O-H Fix Angle
HO  OH  MO  999999999999  52.26  ! H-O-M Fix Angle

DIHEDRALS
!
!V(dihedral) = Kchi(1 + cos(n(chi) - delta))
!
!atom types      Kchi      n      delta

NONBONDED
!
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!
!atom  ignored      epsilon      Rmin/2      ignored      eps,1-4      Rmin/2,1-4
HO      0.000000      0.000000      0.000000      0.0      0.0      0.0
MO      0.000000      0.000000      0.000000      0.0      0.0      0.0
OH      0.000000      -0.18521      1.772873      0.0      0.0      0.0

```

## 17.3 Simulate rigid molecule

Currently, GOMC can simulate rigid molecules for any molecular topology in NVT and NPT ensemble, if none of the Monte Carlo moves that lead to change in molecular configuration (e.g. Regrowth, Crankshaft, IntraSwap, and etc.) was used.

In general, GOMC can simulate rigid molecules in all ensembles for the following molecular topology:

1. Linear and branched molecules with no dihedrals. For instance, carbon dioxide, dimethyl ether, and all water models (SPC, SPC/E, TIP3P, TIP4P, etc).
2. Cyclic molecules, where at least two atoms in all defined angles, belong to the body of the ring. For instance, benzene, toluene, Xylene, and more.

---

### Important:

1. For linear and branched molecule, the molecule's bonds and angles will be adjusted according to the equilibrium values, defined in parameter file.
  2. For cyclic molecules, the molecule's bonds and angles would not change! It is very important to create the initial molecule with correct bonds and angles.
- 

### 17.3.1 Setup rigid molecule

To simulate the rigid molecules in GOMC, we need to perform the following steps:

1. Define all bonds in topology file and use **AUTOGENERATE ANGLES DIHEDRALS** in topology file to specify all angles and dihedral in PSF files.
2. Define all bond parameters in the parameter file. If you wish to not to include the bond energy in your simulation, set the the  $K_b$  to a large value i.e. "999999999999".
3. Define all angle parameters in the parameter file. If you wish to not to include the bend energy in your simulation, set the the  $K_\theta$  to a large value i.e. "999999999999".
4. Define all dihedral parameters in parameter file. If you wish to not to include the dihedral energy in your simulation, set the all the  $C_n$  to zero. **For cyclic molecules only**

## 17.4 Restart the simulation

### 17.4.1 Restart the simulation with Restart

If you intend to start a new simulation from previous simulation state, you can use this option. Restarting the simulation with **Restart** **would not** result in an identical outcome, as if previous simulation was continued. Make sure that in the previous simulation config file, the flag **RestartFreq** was activated and the restart PDB file/files (**OutputName\_BOX\_0\_restart.pdb**) and merged PSF file (**OutputName\_merged.psf**) were printed.

In order to restart the simulation from previous simulation we need to perform the following steps to modify the config file:

1. Set the **Restart** to True.
2. Use the dumped restart PDB file to set the **Coordinates** for each box.
3. Use the dumped merged PSF file to set the **Structure** for both boxes.

4. It is a good practice to comment out the CellBasisVector by adding ‘#’ at the beginning of each cell basis vector. However, GOMC will override the cell basis information with the cell basis data from restart PDB file/files.
5. Use the different OutputName to avoid overwriting the output files.

Here is the example of starting the NPT simulation of dimethyl ether, from equilibrated NVT simulation:

```
#####
# Parameters need to be modified
#####
Restart            true

Coordinates        0    dimethylether_NVT_BOX_0_restart.pdb

Structure          0    dimethylether_NVT_merged.psf

#CellBasisVector1  0      45.00   0.00   0.00
#CellBasisVector2  0      0.00   55.00   0.00
#CellBasisVector3  0      0.00   0.00   45.00

OutputName         dimethylether_NPT
```

Here is the example of starting the NPT-GEMC simulation of dimethyl ether, from equilibrated NVT simulation:

```
#####
# Parameters need to be modified
#####
Restart            true

Coordinates        0    dimethylether_NVT_BOX_0_restart.pdb
Coordinates        1    dimethylether_NVT_BOX_1_restart.pdb

Structure          0    dimethylether_NVT_merged.psf
Structure          1    dimethylether_NVT_merged.psf

#CellBasisVector1  0      45.00   0.00   0.00
#CellBasisVector2  0      0.00   55.00   0.00
#CellBasisVector3  0      0.00   0.00   45.00

#CellBasisVector1  1      45.00   0.00   0.00
#CellBasisVector2  1      0.00   55.00   0.00
#CellBasisVector3  1      0.00   0.00   45.00

OutputName         dimethylether_NPT_GEMC
```

## 17.4.2 Restart the simulation with Checkpoint

If you intend to continue your simulation from previous simulation, you can use this option. Restarting the simulation with Checkpoint would result in an identical outcome, as if previous simulation was continued. Make sure that in the previous simulation config file, the flag `RestartFreq` was activated and the restart PDB file/files (`OutputName_BOX_N_restart.pdb`), restart PSF file/files (`OutputName_BOX_N_restart.psf`), binary coordinate file/files (`OutputName_BOX_N_restart.coor`), XSC file/files (`OutputName_BOX_N_restart.xsc`), and checkpoint file (`OutputName_restart.chk`) were printed.

In order to restart the simulation from previous simulation we need to perform the following steps to modify the config file:

1. Set the Checkpoint to True and provide the Checkpoint file.
2. Use the dumped restart PDB files to set the Coordinates for each box.
3. Use the dumped restart PSF files to set the Structure for both boxes.
4. Use the dumped restart xsc files to set the `extendedSystem` for both boxes.
5. Use the dumped restart coor files to set the `binCoordinates` for both boxes.
6. It is a good practice to comment out the `CellBasisVector` by adding '#' at the beginning of each cell basis vector. However, GOMC will override the cell basis information with the cell basis data from XSC file/files.
7. Use the different `OutputName` to avoid overwriting the output files.

Here is the example of restarting the NPT simulation of dimethyl ether, from equilibrated NVT simulation:

```
#####
# Parameters need to be modified
#####
Checkpoint    true    dimethylether_NVT_restart.chk

Coordinates    0    dimethylether_NVT_BOX_0_restart.pdb

Structure      0    dimethylether_NVT_BOX_0_restart.psf

extendedSystem  0    dimethylether_NVT_BOX_0_restart.xsc

binCoordinates  0    dimethylether_NVT_BOX_0_restart.coor

#CellBasisVector1  0      45.00    0.00    0.00
#CellBasisVector2  0      0.00     55.00    0.00
#CellBasisVector3  0      0.00     0.00    45.00

OutputName      dimethylether_NPT
```

Here is the example of restarting the NPT-GEMC simulation of dimethyl ether, from equilibrated NVT simulation:

```
#####
# Parameters need to be modified
#####
Checkpoint    true    dimethylether_NVT_restart.chk

Coordinates    0    dimethylether_NVT_BOX_0_restart.pdb
Coordinates    1    dimethylether_NVT_BOX_1_restart.pdb
```

(continues on next page)

(continued from previous page)

Structure	0	dimethylether_NVT_BOX_0_restart.psf		
Structure	1	dimethylether_NVT_BOX_1_restart.psf		
extendedSystem	0	dimethylether_NVT_BOX_0_restart.xsc		
extendedSystem	1	dimethylether_NVT_BOX_1_restart.xsc		
binCoordinates	0	dimethylether_NVT_BOX_0_restart.coor		
binCoordinates	1	dimethylether_NVT_BOX_1_restart.coor		
#CellBasisVector1	0	45.00	0.00	0.00
#CellBasisVector2	0	0.00	55.00	0.00
#CellBasisVector3	0	0.00	0.00	45.00
#CellBasisVector1	1	45.00	0.00	0.00
#CellBasisVector2	1	0.00	55.00	0.00
#CellBasisVector3	1	0.00	0.00	45.00
OutputName	dimethylether_NPT_GEMC			

**Note:** As of right now, restarting is not supported for Multi-Sim.

## 17.5 Recalculate the energy

GOMC is capable of recalculate the energy of previous simulation snapshot, with same or different force field. Simulation snapshot is the printed molecule's coordinates at specific steps, which controls by `CoordinatesFreq`. First, we need to make sure that in the previous simulation config file, the flag `CoordinatesFreq` was activated and the coordinates PDB file/files (`OutputName_BOX_0.pdb`) and merged PSF file (`OutputName_merged.psf`) were printed.

In order to recalculate the energy from previous simulation we need to perform the following steps to modify the config file:

1. Set the `Restart` to `True`.
2. Use the dumped coordinates PDB file to set the `Coordinates` for each box.
3. Use the dumped merged PSF file to set the `Structure` for both boxes.
4. Set the `RunSteps` to zero to activate the energy recalculation.
5. Use the different `OutputName` to avoid overwriting the merged PSF files.

**Note:** GOMC only recalculated the energy terms and does not recalculate the thermodynamic properties. Hence, no output file, except merged PSF file, will be generated.

Here is the example of recalculating energy from previous NVT simulation snapshot:

```
#####
# Parameters need to be modified
#####
Restart      true
```

(continues on next page)

(continued from previous page)

Coordinates	0	dimethylether_NVT_BOX_0.pdb
Structure	0	dimethylether_NVT_merged.psf
RunSteps	0	
OutputName		Recalculate

## 17.6 Simulate adsorption

GOMC is capable of simulating gas adsorption in rigid framework using GCMC and NPT-GEMC simulation. In this section, we discuss how to generate PDB and PSF file, how to modify the configuration file to simulate adsorption.

### 17.6.1 Build PDB and PSF file

Generating PDB and PSF file for reservoir is similar to generating PDB and PSF file for isobutane, explained before. Here, we are focusing on how to generate PDB and PSF file for adsorbent. As mentioned before, GOMC can only read PDB and PSF file as input file. If you are using “\*.cif” file for your adsorbent, you need to perform few steps to extend the unit cell and export it as PDB file. There are two ways that you can prepare your adsorption simulation:

#### 1. Using High Throughput Screening (HTS)

GOMC development group created a python code combined with Tcl scripting to automatically generate GOMC input files for adsorption simulation. In this code, we use CoRE-MOF repository created by [Snurr et al.](#) to prepare the simulation input file.

To try this code, execute the following command in your terminal to clone the HTS repository:

```
$ git clone https://github.com/GOMC-WSU/Workshop.git --branch HTS --single-branch
$ cd Workshop
```

or simply download it from [GitHub](#).

Make sure that you installed all [GOMC software requirement](#). Follow the “Readme.md” for more information.

#### 2. Manual Preparation

To illustrate the steps that need to be taken to prepare the PDB and PSF file, we will use an example provided in one of our workshop. Make sure that you installed all [GOMC software requirement](#).

To clone the workshop, execute the following command in your terminal to clone the workshop:

```
$ git clone https://github.com/GOMC-WSU/Workshop.git --branch master --single-branch
```

or simply download it from [GitHub](#).

To show how to extend the unit cell of IRMOF-1 and build the PDB and PSF files, change your directory to:

```
$ cd Workshop/adsorption/GCMC/argon_IRMOF_1/build/base/.
```

In this directory, there is a “README.txt” file, which provides detailed information of steps need to be taken. Here we just provide a summary of these steps:

- Extend the unit cell of “EDUSIF\_clean\_min.cif” file using [VESTA](#). To learn how to extend the unit cell, removing bonds, and export it as PDB file, please refer to this [document](#) to generate “EDUSIF\_clean\_min.pdb” file.

---

**Note:** Generated PDB file does not provide all necessary information. Further modification must be made.

---

- The easy way to generate PSF file is to treat each atom as a separate molecule kind to avoid defining bonds, angles, and dihedrals. To modify the “EDUSIF\_clean\_min.pdb” file (set the residue ID, resname, ...), execute the following command to generate the “EDUSIF\_clean\_min\_modified.pdb” file.

```
vmd -dispdev text < convert_VESTA_PDB.tcl
```

- Treating each atom as separate molecule kind will make it easy to generate topology file. Here is an example of topology file where each atom is treated as a separate residue kind:

```
* Topology file for IRMOF-1 (Zn4O(BDC)3)
!
MASS 1 O 15.999 O !
MASS 2 C 12.011 C !
MASS 3 H 1.008 H !
MASS 4 ZN 65.380 ZN !

DEFA FIRS none LAST none
AUTOGENERATE ANGLES DIHEDRALS

RESI C 0.000
GROUP
ATOM C C 0.000
PATCHING FIRS NONE LAST NONE

RESI H 0.000
GROUP
ATOM H H 0.000
PATCHING FIRS NONE LAST NONE

RESI O 0.000
GROUP
ATOM O O 0.000
PATCHING FIRS NONE LAST NONE

RESI Zn 0.000
GROUP
ATOM Zn ZN 0.000
PATCHING FIRS NONE LAST NONE

END
```

- To generate the PSF file, each molecule kind must be separated and stored in separate pdb file. Then we use VMD to generate the PSF file. All these process are scripted in “build\_EDUSIF\_auto.tcl” and we just need to execute the following command to generate the “IRMOF\_1\_BOX\_0.pdb” and “IRMOF\_1\_BOX\_0.psf” files.

```
vmd -dispdev text < build_EDUSIF_auto.tcl
```

- Last steps to fix the adsorbent atoms in their position. As mentioned in PDB section, setting the Beta = 1.00 value of a molecule in PDB file, will fix that molecule position. This can be done by a text editor but here we use another Tcl scrip to do that. Execute the following command in your terminal to set the Beta value of all atoms in “IRMOF\_1\_BOX\_0.pdb” to 1.00.

```
vmd -dispdev text < setBeta.tcl
```

## 17.6.2 Adsorption in GCMC

To simulate adsorption using GCMC ensemble, we need to perform the following steps to modify the config file:

1. Use the generated PDB files for adsorbent and adsorbate to set the Coordinates.
2. Use the generated PSF files for adsorbent and adsorbate to set the Structure.
3. Calculate the cell basis vectors for each box and set the CellBasisVector1,2,3 for each box.

**Note:** To calculate the cell basis vector with cell length  $a, b, c$  and cell angle  $\alpha, \beta, \gamma$  we use the following equations:

$$a_x = a$$

$$a_y = 0.0$$

$$a_z = 0.0$$

$$b_x = b \times \cos(\gamma)$$

$$b_y = b \times \sin(\gamma)$$

$$c_x = c \times \cos(\beta)$$

$$c_y = c \times \frac{\cos(\alpha) - \cos(\beta) \times \cos(\gamma)}{\sin(\gamma)}$$

$$c_z = c \times \sqrt{\sin(\beta)^2 - \left( \frac{\cos(\alpha) - \cos(\beta) \times \cos(\gamma)}{\sin(\gamma)} \right)^2}$$

$$\text{CellBasisVector1} = (a_x, a_y, a_z)$$

$$\text{CellBasisVector2} = (b_x, b_y, b_z)$$

$$\text{CellBasisVector3} = (c_x, c_y, c_z)$$

4. Set the Fugacity for adsorbate and include Fugacity for adsorbent with arbitrary value (e.g. 0.00).

Here is the example of argon (AR) adsorption at 5 bar in IRMOF-1 using GCMC ensemble:

```
#####
# Parameters need to be modified
#####
Coordinates    0    ../build/base/IRMOF_1_BOX_0.pdb
Coordinates    1    ../build/reservoir/START_BOX_1.pdb

Structure      0    ../build/base/IRMOF_1_BOX_0.psf
Structure      1    ../build/reservoir/START_BOX_1.psf
```

(continues on next page)



(continued from previous page)

CellBasisVector1	0	36.8140	0.00	0.00
CellBasisVector2	0	18.2583	31.9880	0.00
CellBasisVector3	0	18.2712	10.5596	30.1748
CellBasisVector1	1	40.00	0.00	0.00
CellBasisVector2	1	0.00	40.00	0.00
CellBasisVector3	1	0.00	00.00	40.00
Fugacity	AR	5.0		
Fugacity	C	0.0		
Fugacity	H	0.0		
Fugacity	O	0.0		
Fugacity	ZN	0.0		

### 17.6.3 Adsorption in NPT-GEMC

To simulate adsorption using NPT-GEMC ensemble, simulation box 0 is used for adsorbent with fixed volume and simulation box 1 is used for adsorbate, where volume of this box is fluctuating at imposed pressure. To simulation adsorption in NPT-GEMC ensemble we need to perform the following steps to modify the config file:

1. Use the generated PDB file for adsorbent to set the Coordinates for box 0.
2. Use the generated PDB file for adsorbate to set the Coordinates for box 1.
3. Use the generated PSF file for adsorbent to set the Structure for box 0.
4. Use the generated PSF file for adsorbate to set the Structure for box 1.
5. Calculate the cell basis vectors for each box and set the CellBasisVector1,2,3 for each box.
6. Set the GEMC simulation type to "NPT".
7. Set the imposed Pressure (bar) for fluid phase.
8. Keep the volume of box 0 constant by activating the FixVolBox0.

Here is the example of argon (AR) adsorption at 5 bar in IRMOF-1 using NPT-GEMC ensemble:

```
#####
# Parameters need to be modified
#####
Coordinates      0    ../build/base/IRMOF_1_BOX_0.pdb
Coordinates      1    ../build/reservoir/START_BOX_1.pdb

Structure        0    ../build/base/IRMOF_1_BOX_0.psf
Structure        1    ../build/reservoir/START_BOX_1.psf

CellBasisVector1  0    36.8140    0.00    0.00
CellBasisVector2  0    18.2583    31.9880    0.00
CellBasisVector3  0    18.2712    10.5596    30.1748

CellBasisVector1  1    40.00    0.00    0.00
CellBasisVector2  1    0.00    40.00    0.00
CellBasisVector3  1    0.00    00.00    40.00
```

(continues on next page)

(continued from previous page)

```
GEMC          NPT
Pressure      5.0
FixVolBox0    true
```

## 17.7 Calculate Solvation Free Energy

GOMC is capable of calculating absolute solvation free energy in NVT or NPT ensemble. Here we are focusing how to setup the GOMC simulation files to calculate absolute solvation free energy.

GOMC outputs the required informations ( $\frac{dE_\lambda}{d\lambda}$ ,  $\Delta E_\lambda$ ) to calculate solvation free energy with various estimators, such as TI, BAR, MBAR, and more.

### 17.7.1 Setup Simulation Files

#### 1. Using FreeEnergy BASH Script

GOMC development group created a BASH script combined with Tcl scripting to automatically generate GOMC input files for free energy simulations in NVT (master branch) or NPT (NPT branch) ensemble.

To try this script, execute the following command in your terminal to clone the FreeEnergy repository:

```
$ git clone https://github.com/msorouseh/FreeEnergy.git
$ cd FreeEnergy
```

or simply download it from [GitHub](https://github.com/msorouseh/FreeEnergy).

Make sure that you installed all [GOMC software requirement](#). Follow the [README](#) for more information.

#### 2. Manual Preparation

To simulate solvation free energy, we need to perform the following steps:

- Generate the PDB and PSF files for a system contains 1 solute +  $N$  solvent molecules.

---

**Note:** Number of solvent molecules ( $N$ ) must be determined by user, based on the system size.

---

- Equilibrate your system in NVT ensemble at specified **Temperature**.
- Equilibrate your system in NPT ensemble at specified **Temperature** and **Pressure**, using PDB and PSF **restart** files, generated from previous equilibration simulation.
- Determine the number of intermediate states that lead to adequate overlaps between neighboring states.
- For each intermediate state ( $\lambda_i$ ), create an unique directory and perform the following steps:
  1. Use the **restart** PDB file, generated from NPT equilibration simulation, to set the **Coordinates**.
  2. Use the merged PSF files, generated from NPT equilibration simulation, to set the **Structure**.
  3. Define the free energy parameters in **config** file:
    - Set the frequency of free energy calculation
    - Set the solute molecule kind name (resname) and number (resid)

- Set the soft-core parameters
- Define the lambda vecotrs for VDW and Coulomb interaction
- Set the index ( $i$ ) of the lambda vetor ( $\lambda$ ), at which solute-solvent interaction will be coupled with  $\lambda_i$ , using `InitialState` keyword.

Here is the example of free energy parameters for CO2 (resid 1) solvation, with 9 intermediate states, where the solute-solvent interaction will be coupled with  $\lambda_{\text{VDW}}(6) = 1.0$ ,  $\lambda_{\text{Elect}}(6) = 0.50$ .

```
#####
# FREE ENERGY PARAMETERS
#####
FreeEnergyCalc true    1000
MoleculeType  CO2    1
InitialState    6
ScalePower      2
ScaleAlpha      0.5
MinSigma        3.0
ScaleCoulomb    false
#states         0      1      2      3      4      5      6      7      8
LambdaVDW       0.00  0.25  0.50  0.75  1.00  1.00  1.00  1.00  1.00
LambdaCoulomb   0.00  0.00  0.00  0.00  0.00  0.25  0.50  0.75  1.00
```

4. Equilibrate your system in NVT or NPT ensemble.
5. Perform the production simulation in NVT or NPT ensemble.

## 17.7.2 Process GOMC Free Energy Outputs

I free energy perturbation method, the free energy difference between two states A ( $\lambda = 0.0$ ) and B ( $\lambda = 1.0$ ), with N - 2 intermediate states is given by:

$$\Delta G(A \rightarrow B) = -\frac{1}{\beta} \sum_{i=0}^{N-1} \ln \langle \exp(-\beta \Delta E_{i,i+1}) \rangle_i$$

where  $\Delta E_{i,i+1} = E_{i+1} - E_i$  is the energy difference of the system between states  $i$  and  $i+1$ , and  $\langle \rangle_i$  is the ensemble average for simulation performed in intermediate state  $i$ .

In thermodynamic integration, the free energy change is calculated from

$$\Delta G(A \rightarrow B) = \int_{\lambda=0}^{\lambda=1} \langle \frac{dU_{\lambda}}{d\lambda} \rangle_{\lambda} d\lambda$$

where  $\frac{dU_{\lambda}}{d\lambda}$  is the derivative of energy with respect to  $\lambda$ , and  $\langle \rangle_{\lambda}$  is the ensemble average for a simulation run at intermediate state  $\lambda$ .

GOMC outputs the raw informations, such as the lambda intermediate states, the derivative of energy with respect to current lambda ( $\frac{dE_{\lambda}}{d\lambda}$ ), the energy different between current lambda state and all other neighboring lambda states ( $\Delta E_{\lambda}$ ), which is essential to calculate solvation free energy with various estimators, such as TI, BAR, MBAR, and more.

There are variety of tools developed to caculate free energy difference, including [alchemlyb](#) and [alchemical-analysis](#).

### 1. Alchemlyb

In [alchemlyb](#), a variety of methods can be used to estimate the free energy, including thermodynamic integration (TI), Bennett acceptance ratio (BAR), and multistate Bennett acceptance ratio (MBAR). [alchemlyb](#) is also capable of loading GOMC free energy output files (Free\_Energy\_BOX\_0\_OutputName.dat).

```
#T = 298.0000(K), Lambda State 6: (lambda Coulomb, lambda VDW) = (0.5000,1.0000)
#Steps      Total_En(kJ/mol)      dU/dL(Coulomb=0.5000)      dU/dL(VDW=1.0000)  DelE(L->(0.00\
00,0.0000)) DelE(L->(0.0000,0.2500)) DelE(L->(0.0000,0.5000)) DelE(L->(0.0000,0.7500)) DelE(L->(0.00\
00,1.0000)) DelE(L->(0.2500,1.0000)) DelE(L->(0.5000,1.0000)) DelE(L->(0.7500,1.0000)) DelE(L->(1.00\
00,1.0000))
```

Fig. 1: Snapshot of GOMC free energy output file (Free\_Energy\_BOX\_0\_OutputName.dat).

To learn more about alchemlybe, please refer to [alchemlyb documentation](#) or [alchemlyb GitHub](#) page.

---

**Note:** Currently, alchemlyb does not support the free energy plots, overlap analysis, and free energy convergence analysis.

---

To use this tool, you must install python 3 and then execute the following command in your terminal to install alchemlyb:

```
$ pip install alchemlyb
```

## 2. Alchemical Analysis

The alchemical-analysis tools is developed by Mobley group at MIT, to Analyze alchemical free energy calculations conducted in GROMACS, AMBER or SIRE. Alchemical Analysis is still available but deprecated and in the process of migrating all functionality to [alchemlyb](#) tool.

Alchemical Analysis tool handles analysis via a slate of free energy methods, including BAR, MBAR, TI, and the Zwanzig relationship (exponential averaging) among others, and provides a good deal of analysis of computed free energies and convergence in order to help you assess the quality of your results.

Since alchemical-analysis is no longer supported by its developers, the GOMC parser for this tool was implemented and stored in a separate [repository](#).

---

**Note:** We encourage user to use [alchemlyb GitHub](#) tools for plotting, once all the plotting features and free energy analysis was migrated.

---

To use this tool, you must install python 2 and then execute the following command in your terminal to clone the alchemical-analysis repository:

```
$ git clone https://github.com/msorosh/alchemical-analysis.git
$ cd alchemical-analysis
$ sudo python setup.py install
```

## 17.8 Run a Multi-Sim

GOMC can automatically generate independent simulations with varying temperatures from one input file. This allows the user to sample a wider search space. To do so GOMC must be compiled in MPI mode, and a couple of parameters must be added to the conf file.

To compile in MPI mode, navigate to the GOMC/ directory and issue the following commands:

```
$ chmod u+x metamakeMPI.sh
$ ./metamakeMPI.sh
```

Then once the compilation is complete, set up the conf file as you would for a standard GOMC simulation.

Finally, enter more than one value for Temperature separated by a tab or space.

```
#####
# SIMULATION CONDITION
#####
Temperature    270.00    280.00    290.00    300.00
```

A folder will be created for the output of each simulation, and the name will be generated from the temperatures you choose. A parent folder containing all the child folders will be created so as to not overpopulate the initial directory. You may elect to choose the name of the folder in which all the sub-folders for each replica are contained. Enter this name as a string following the `MultiSimFolderName` parameter. If you don't provide this parameter, the default "MultiSimFolderName" will be used.

```
MultiSimFolderName  outputFolderName
```

**Note:** To perform a multisim, GOMC must be compiled in MPI mode. Also, if GOMC is compiled in MPI mode, a multisim must be performed. To perform a standard simulation, use standard GOMC.

The rest of the conf file should be similar to how you would set up a standard GOMC simulation.

To initiate the multi-sim, first decide how many MPI processes and openMP threads you want to use and call GOMC with the following format.

```
$ mpiexec -n #ofsimulations GOMC_xxx_yyyy +p<#ofthreads>(optional) conffile
```

The number of MPI processes must equal the number of simulations you wish to run. Each will by default be assigned one openMP thread; however, if you have leftover processors, you may assign them as openMP threads. There must be an equal amount of openMP threads assigned to each process.

A formula to determine how many threads to use is as follows:

$$OpenMPThreads = \text{floor}[(NumberOfProcessorsAvailable - NumberOfMPIProcesses)/NumberOfMPIProcesses]$$

Floor[ ] - Rounds down a real number to the nearest integer.

For example, if I have 7 processors and I wanted to run 2 simulations in my multi-sim.

$$OpenMPThreads = \text{floor}[(7 - 2)/2] = \text{floor}[2.5] = 2$$

```
$ mpiexec -n 2 ./GOMC_CPU_GEMC +p2 in.conf
```



## GET HELP OR TECHNICAL SUPPORT

For get any help or technical support, please send message to GOMC gitter:

[https://gitter.im/GOMC\\_WSU/Lobby](https://gitter.im/GOMC_WSU/Lobby)

or send email to:

- Jeffrey Potoff: [jpotoff@wayne.edu](mailto:jpotoff@wayne.edu)
- Loren Schwiebert: [loren@wayne.edu](mailto:loren@wayne.edu)