

Foundation of Multiscale Modelling: Kinetics

Seminar/Lab 3
Introduction in molecular dynamics.
Practice in LAMMPS and VMD.

Alexey Tsukanov, a.tsukanov@skoltech.ru

Outline

- Aims of the lesson
- Installation of LAMMPS
- Molecular dynamics model (case: molecular system)
- LAMMPS + VMD
- Example of input-file for LAMMPS
- HW3, tasks 1-4.

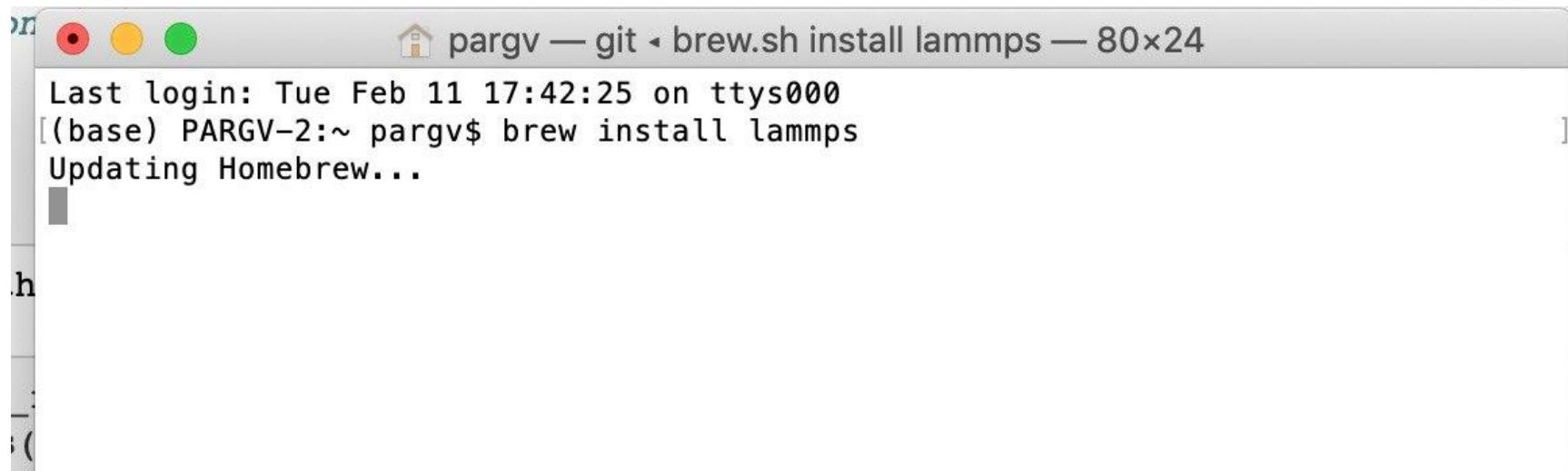
Aims of the lesson and HW 3

- ✓ Understanding of molecular (not “atomic”) systems
 - ✓ Bonds
 - ✓ Angles
 - ✓ Charges
- ✓ Numerical study of system of non-spherical / dipolar molecules and LJ
 - ✓ Orientational autocorrelation function $K_{\theta}(t)$ and $\Psi_{\theta,\parallel}(t)$ of water
 - ✓ Response of $K_{\theta}(t)$ of water on external electric field
 - ✓ Mean square displacement $MSD(t)$ and diffusion coefficient
 - ✓ VACF(t) and diffusion coefficient (LJ case)
 - ✓ Pair distribution functions, hydrogen bonding
 - ✓ Modeling of different aggregate states of mater (LJ case)
- ✓ Getting practical skills:
 - ✓ Practice in LAMMPS and LAMMPS + VMD
 - ✓ Preparation / generation of molecular system
 - ✓ Simulation in NVT, NpT conditions
 - ✓ Save / dump results
 - ✓ Analysis and visualization

Installation of LAMMPS

Case 1: MacOS

```
brew install lammps
```



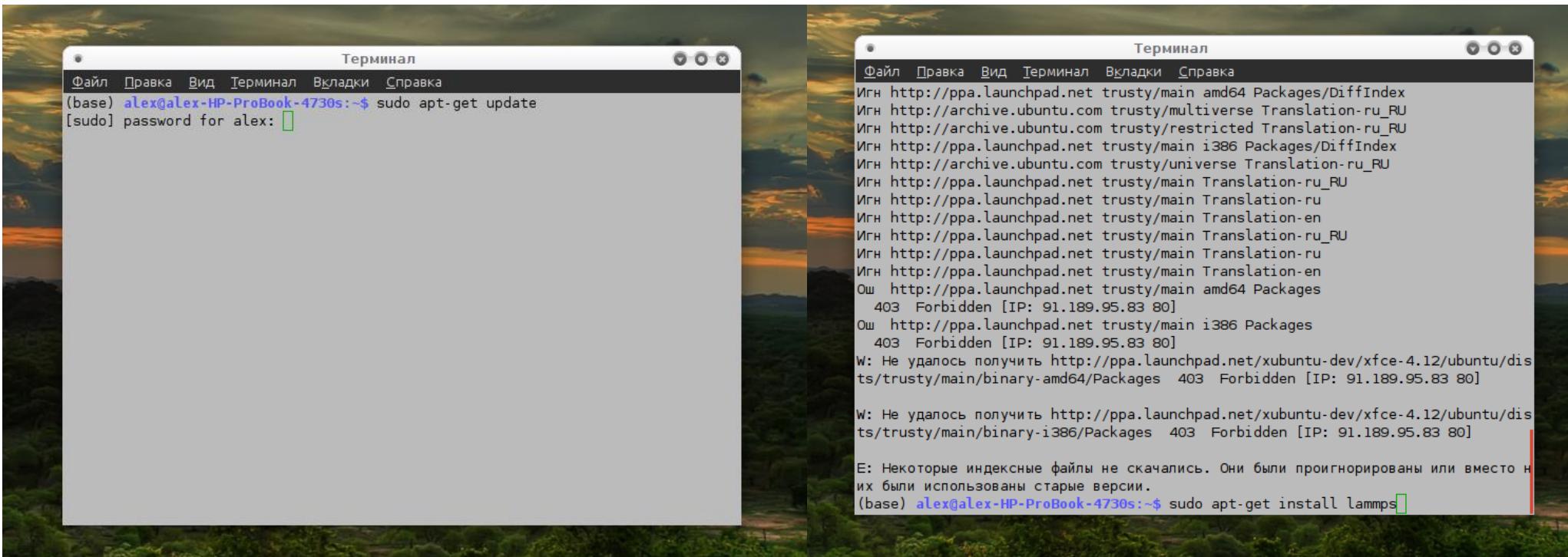
A screenshot of a macOS terminal window titled "pargv — git - brew.sh install lammps — 80x24". The window shows the command "brew install lammps" being run, along with the output of the command, which includes "Updating Homebrew...".

```
Last login: Tue Feb 11 17:42:25 on ttys000
(base) PARGV-2:~ pargv$ brew install lammps
Updating Homebrew...
```

Installation of LAMMPS

Case 2: Linux (via apt-get)

```
sudo apt-get update  
sudo apt-get install lammps
```



```
(base) alex@alex-HP-ProBook-4730s:~$ sudo apt-get update  
[sudo] password for alex:  
  
Файл Правка Вид Терминал Вкладки Справка  
Игн http://ppa.launchpad.net trusty/main amd64 Packages/DiffIndex  
Игн http://archive.ubuntu.com trusty/multiverse Translation-ru_RU  
Игн http://archive.ubuntu.com trusty/restricted Translation-ru_RU  
Игн http://ppa.launchpad.net trusty/main i386 Packages/DiffIndex  
Игн http://archive.ubuntu.com trusty/universe Translation-ru_RU  
Игн http://ppa.launchpad.net trusty/main Translation-ru_RU  
Игн http://ppa.launchpad.net trusty/main Translation-ru  
Игн http://ppa.launchpad.net trusty/main Translation-en  
Игн http://ppa.launchpad.net trusty/main Translation-ru_RU  
Игн http://ppa.launchpad.net trusty/main Translation-ru  
Игн http://ppa.launchpad.net trusty/main Translation-en  
Ош http://ppa.launchpad.net trusty/main amd64 Packages  
    403 Forbidden [IP: 91.189.95.83 80]  
Ош http://ppa.launchpad.net trusty/main i386 Packages  
    403 Forbidden [IP: 91.189.95.83 80]  
W: Не удалось получить http://ppa.launchpad.net/xubuntu-dev/xfce-4.12/ubuntu/dis  
ts/trusty/main/binary-amd64/Packages 403 Forbidden [IP: 91.189.95.83 80]  
W: Не удалось получить http://ppa.launchpad.net/xubuntu-dev/xfce-4.12/ubuntu/dis  
ts/trusty/main/binary-i386/Packages 403 Forbidden [IP: 91.189.95.83 80]  
E: Некоторые индексные файлы не скачались. Они были проигнорированы или вместо н  
их были использованы старые версии.  
(base) alex@alex-HP-ProBook-4730s:~$ sudo apt-get install lammps
```

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page

The screenshot shows a web browser displaying the LAMMPS Molecular Dynamics Simulator website. The URL in the address bar is lammps.sandia.gov. The page title is "LAMMPS Molecular Dynamics Simulator". A dictionary definition of "lamp" is shown: "a device that generates light, heat, or therapeutic radiation; something that illuminates the mind or soul -- www.dictionary.com". Below the definition is a molecular simulation visualization with a caption: "hover to animate -- [input script](#)". The visualization shows a grid of blue and yellow spheres. Below the visualization is a caption: "physical analog (start at 3:25) & [explanation](#)". The main content area features a 5x7 grid of links:

Big Picture	Code	Documentation	Results	Related Tools	Context	User Support
Features	Download	Manual	Publications	Pre/Post processing	Authors	Mail list
Non-features	GitHub	Developer guide	Pictures	Pizza.py Toolkit	History	IRC channel
Packages	SourceForge	Tutorials	Movies	Offsite LAMMPS packages & tools	Funding	Workshops
FAQ	Latest features & bug fixes	MD to LAMMPS glossary	Benchmarks	Visualization	Open source	Contribute to LAMMPS
Wish list	Report bugs & request features	Commands	Citing LAMMPS	Related modeling codes		

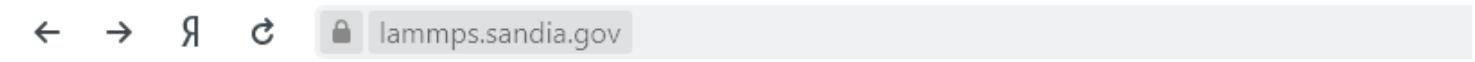
Below the grid are five thumbnail images of scientific journals: Polymer Chemistry, THE JOURNAL OF PHYSICAL CHEMISTRY C, THE JOURNAL OF PHYSICAL CHEMISTRY A, PHYSICAL REVIEW LETTERS, and Philosophical Magazine.

At the bottom of the page, a footer states: "LAMMPS is a classical molecular dynamics code with a focus on materials modeling. It's an acronym for Large-scale molecular simulator." followed by the URL <https://lammps.sandia.gov/#nogo>.

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS



Download LAMMPS

You can download LAMMPS as a tarball from this page, using the "Download Now" button below.

There are several ways to get the LAMMPS software, either as a tarball, or from an active repository, or in executable form:

- [Download a tarball \(here or from GitHub\)](#)
- [Git repository for LAMMPS](#)
- [SVN repository for LAMMPS](#)
- [Pre-built Linux executables](#)
- [Pre-built Mac executables](#)
- [Pre-built Windows executables](#)

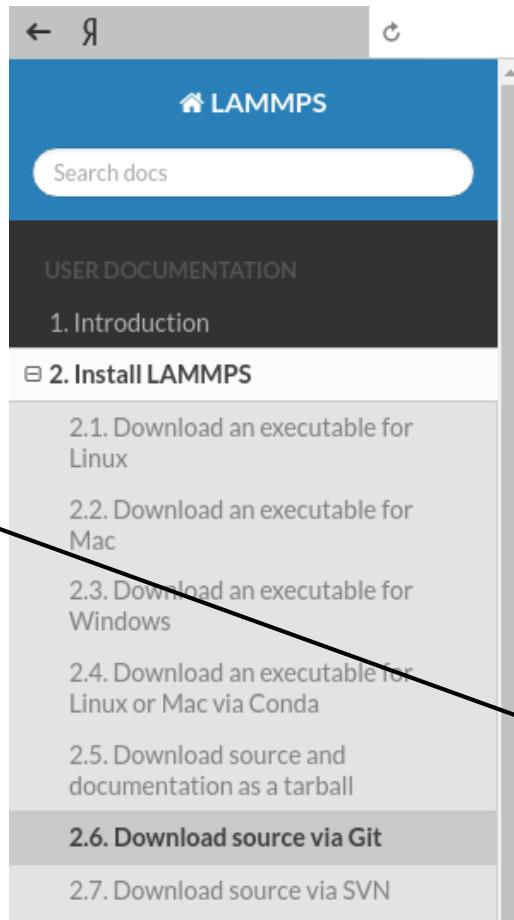
With source code, you have to [build LAMMPS](#) using "cmake" or "make". But you have more flexibility as to what features to include.

The [Install](#) doc page lists what is included in the LAMMPS distribution.

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS
4. Copy string



lammps.sandia.gov 2.6. Download source via Git

You must have [Git](#) installed on your system to communicate with the public Git repository.

Warning

As of Oct 2016, the official home of public LAMMPS development is on GitHub. The LAMMPS website is deprecated, may not be up-to-date, and may go away at any time.

You can follow LAMMPS development on 3 different Git branches:

- **stable** : this branch is updated with every stable release
- **unstable** : this branch is updated with every patch release
- **master** : this branch continuously follows ongoing development

To access the Git repositories on your box, use the clone command to create a local copy:

```
git clone -b unstable https://github.com/lammps/lammps.git mylammps
```

where "mylammps" is the name of the directory you wish to create on your machine. You can switch between branches; you can switch between them at any time using "git checkout <bran...

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS
4. Copy string and past to terminal
5. Change branch:
 - ~~-b unstable~~
 - ~~-b stable~~

The screenshot shows a terminal window titled "Терминал" (Terminal) in Russian. The window displays the following text:

```
Preparing to unpack .../liberror-perl_0.17-1.1_all.deb ...
Unpacking liberror-perl (0.17-1.1) ...
Выбор ранее не выбранного пакета git-man.
Preparing to unpack .../git-man_1%3a1.9.1-1ubuntu0.10_all.deb ...
Unpacking git-man (1:1.9.1-1ubuntu0.10) ...
Выбор ранее не выбранного пакета git.
Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.10_amd64.deb ...
Unpacking git (1:1.9.1-1ubuntu0.10) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Настраивается пакет liberror-perl (0.17-1.1) ...
Настраивается пакет git-man (1:1.9.1-1ubuntu0.10) ...
Настраивается пакет git (1:1.9.1-1ubuntu0.10) ...
localepurge: Disk space freed in /usr/share/locale: 0 KiB
localepurge: Disk space freed in /usr/share/man: 0 KiB
localepurge: Disk space freed in /usr/share/gnome/help: 0 KiB
localepurge: Disk space freed in /usr/share/omf: 0 KiB
localepurge: Disk space freed in /usr/share/doc/kde/HTML: 0 KiB

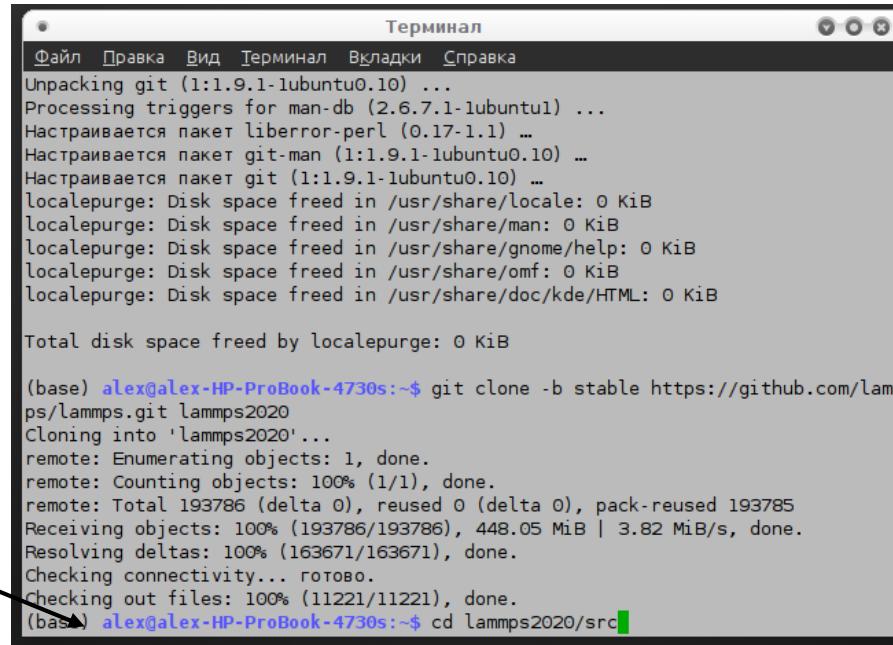
Total disk space freed by localepurge: 0 KiB
(base) alex@alex-HP-ProBook-4730s:~$ git clone -b stable https://github.com/lammps/lammps.git lammps2020
Cloning into 'lammps2020'...
```

```
git clone -b stable https://github.com/lammps/lammps.git lammps2020
```

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS
4. Copy string and past to terminal
5. Change branch: –b stable
6. Go to lammps /src folder



```
Терминал
Файл Правка Вид Терминал Вкладки Справка
Unpacking git (1:1.9.1-1ubuntu0.10) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Настраивается пакет liberror-perl (0.17-1.1) ...
Настраивается пакет git-man (1:1.9.1-1ubuntu0.10) ...
Настраивается пакет git (1:1.9.1-1ubuntu0.10) ...
localepurge: Disk space freed in /usr/share/locale: 0 KiB
localepurge: Disk space freed in /usr/share/man: 0 KiB
localepurge: Disk space freed in /usr/share/gnome/help: 0 KiB
localepurge: Disk space freed in /usr/share/omf: 0 KiB
localepurge: Disk space freed in /usr/share/doc/kde/HTML: 0 KiB

Total disk space freed by localepurge: 0 KiB

(base) alex@alex-HP-ProBook-4730s:~$ git clone -b stable https://github.com/lammps/lammps.git lammps2020
Cloning into 'lammps2020'...
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 193786 (delta 0), reused 0 (delta 0), pack-reused 193785
Receiving objects: 100% (193786/193786), 448.05 MiB | 3.82 MiB/s, done.
Resolving deltas: 100% (163671/163671), done.
Checking connectivity... готово.
Checking out files: 100% (11221/11221), done.
(base) alex@alex-HP-ProBook-4730s:~$ cd lammps2020/src
```

cd lammps2020/src

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
 2. Go to Download page
 3. Go to Git repository for LAMMPS
 4. Copy string and past to terminal
 5. Change branch: –b stable
 6. Go to lammmps /src folder

**It contains source files for
commands, computes, fixes etc >>**

as well as packages (in blue)

Файл	Правка	Вид	Терминал	Вкладки	Справка	
atom_vec.cpp				compute_stress_atom.cpp	fix_nph.h	Make.sh
atom_vec_ellipsoid.cpp				compute_stress_atom.h	fix_nph_sphere.cpp	MANYBODY
atom_vec_ellipsoid.h				compute_temp_chunk.cpp	fix_nph_sphere.h	math_complex.h
atom_vec.h				compute_temp_chunk.h	fix_npt.cpp	math_const.h
atom_vec_hybrid.cpp				compute_temp_com.cpp	fix_npt.h	math_extra.cpp
atom_vec_hybrid.h				compute_temp_com.h	fix_npt_sphere.cpp	math_extra.h
atom_vec_line.cpp				compute_temp.cpp	fix_npt_sphere.h	math_special.cpp
atom_vec_line.h				compute_temp_deform.cpp	fix_nve.cpp	math_special.h
atom_vec_sphere.cpp				compute_temp_deform.h	fix_nve.h	math_vector.h
atom_vec_sphere.h				compute_temp.h	fix_nve_limit.cpp	MC
atom_vec_tri.cpp				compute_temp_partial.cpp	fix_nve_limit.h	memory.cpp
atom_vec_tri.h				compute_temp_partial.h	fix_nve_noforce.cpp	memory.h
balance.cpp				compute_temp_profile.cpp	fix_nve_noforce.h	mergesort.h
balance.h				compute_temp_profile.h	fix_nve_sphere.cpp	MESSAGE
BODY				compute_temp_ramp.cpp	fix_nve_sphere.h	min_cg.cpp
body.cpp				compute_temp_ramp.h	fix_nvt.cpp	min_cg.h
body.h				compute_temp_region.cpp	fix_nvt.h	min.cpp
bond.cpp				compute_temp_region.h	fix_nvt_sllod.cpp	min_fire.cpp
bond_DEPRECATED.cpp				compute_temp_sphere.cpp	fix_nvt_sllod.h	min_fire.h
bond_DEPRECATED.h				compute_temp_sphere.h	fix_nvt_sphere.cpp	min.h
bond.h				compute_torque_chunk.cpp	fix_nvt_sphere.h	min_hftn.cpp
bond_hybrid.cpp				compute_torque_chunk.h	fix_planeforce.cpp	min_hftn.h
bond_hybrid.h				compute_vacf.cpp	fix_planeforce.h	minimize.cpp
bond_zero.cpp				compute_vacf.h	fix_press_berendsen.cpp	minimize.h
bond_zero.h				compute_vcm_chunk.cpp	fix_press_berendsen.h	min_linesearch.cpp
change_box.cpp				compute_vcm_chunk.h	fix_print.cpp	min_linesearch.h
change_box.h				CORESHELL	fix_print.h	min_quickmin.cpp
citeme.cpp				create_atoms.cpp	fix_property_atom.cpp	min_quickmin.h
citeme.h				create_atoms.h	fix_property_atom.h	min_sd.cpp
CLASS2				create_bonds.cpp	fix_read_restart.cpp	min_sd.h
COLLOID				create_bonds.h	fix_read_restart.h	MISC
comm_brick.cpp				create_box.cpp	fix_recenter.cpp	modify.cpp
comm_brick.h				create_box.h	fix_recenter.h	modify.h
comm.cpp				delete_atoms.cpp	fix_respa.cpp	MOLECULE
comm.h				delete_atoms.h	fix_respa.h	molecule.cpp
comm_tiled.cpp				delete_bonds.cpp	fix_restrain.cpp	molecule.h
comm_tiled.h				delete_bonds.h	fix_restrain.h	MPIIO
COMPRESS				DEPEND	fix_setforce.cpp	mpio.h
compute_adf.cpp				Depend.sh	fix_setforce.h	MSCG
compute_adf.h				deprecated.cpp	fix_spring_chunk.cpp	my_page.h
compute_aggregate_atom.cpp				deprecated.h	fix_spring_chunk.h	my_pool_chunk.h
compute_aggregate_atom.h				dihedral.cpp	fix_spring.cpp	nbin.cpp
compute_angle.cpp				dihedral_DEPRECATED.cpp	fix_spring.h	nbin.h
compute_angle.h				dihedral_DEPRECATED.h	fix_spring_rg.cpp	nbin_standard.cpp

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS
4. Copy string and past to terminal
5. Change branch: –b stable
6. Go to lammmps /src folder
7. make clean-all

make clean-all

```
compute_improper_local.h          fix.cpp           improper.cpp
compute_inertia_chunk.cpp         fix_deform.cpp   improper_DEPRECATED.cpp
compute_inertia_chunk.h          fix_deform.h     improper_DEPRECATED.h
compute_ke_atom.cpp              fix_DEPRECATED.cpp
compute_ke_atom.h                fix_DEPRECATED.h
compute_ke.cpp                  fix_drag.cpp      improper.H
compute_ke.h                   fix_drag.h       improper_hybrid.cpp
compute_msd_chunk.cpp            fix_dt_reset.cpp  improper_hybrid.h
compute_msd_chunk.h              fix_dt_reset.h   improper_zero.cpp
compute_msd.cpp                 fix_enforce2d.cpp
compute_msd.h                   fix_enforce2d.h
compute_omega_chunk.cpp          fix_external.cpp
compute_omega_chunk.h            fix_external.h
compute_orientorder_atom.cpp     fix_gravity.cpp
compute_orientorder_atom.h       fix_gravity.h
compute_pair.cpp                fix_group.cpp
compute_pair.h                  fix_group.h
compute_pair_local.cpp           fix.h
compute_pair_local.h             fix_halt.cpp
(base) alex@alex-HP-ProBook-4730s:~/Lammps2020/src$ make clean-all
rm -rf Obj_*
(base) alex@alex-HP-ProBook-4730s:~/Lammps2020/src$
```



Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS
4. Copy string and past to terminal
5. Change branch: –b stable
6. Go to lammmps /src folder
7. make clean-all
8. Add packages:

```
make yes-molecule  
make yes-kspace
```

*Allows modelling
of molecular
systems*

*Allows calculation
of long-range
electrostatics
(Ewald, PPPM)*

The screenshot shows a terminal window with a dark theme. At the top, there is a menu bar with Russian labels: Файл (File), Правка (Edit), Вид (View), Терминал (Terminal), Вкладки (Tabs), and Справка (Help). Below the menu, the terminal displays a list of files in the LAMMPS source directory, including various C++ and header files for different compute, fix, and group classes. The list is color-coded by package: GPU (blue), GRANULAR (red), KIM (green), and KOKKOS (orange). Arrows from the text boxes in the previous slide point to specific files in this list: 'molecule.cpp' and 'kspace.cpp' are highlighted in blue, while 'pair.cpp' and 'pair_local.cpp' are highlighted in green.

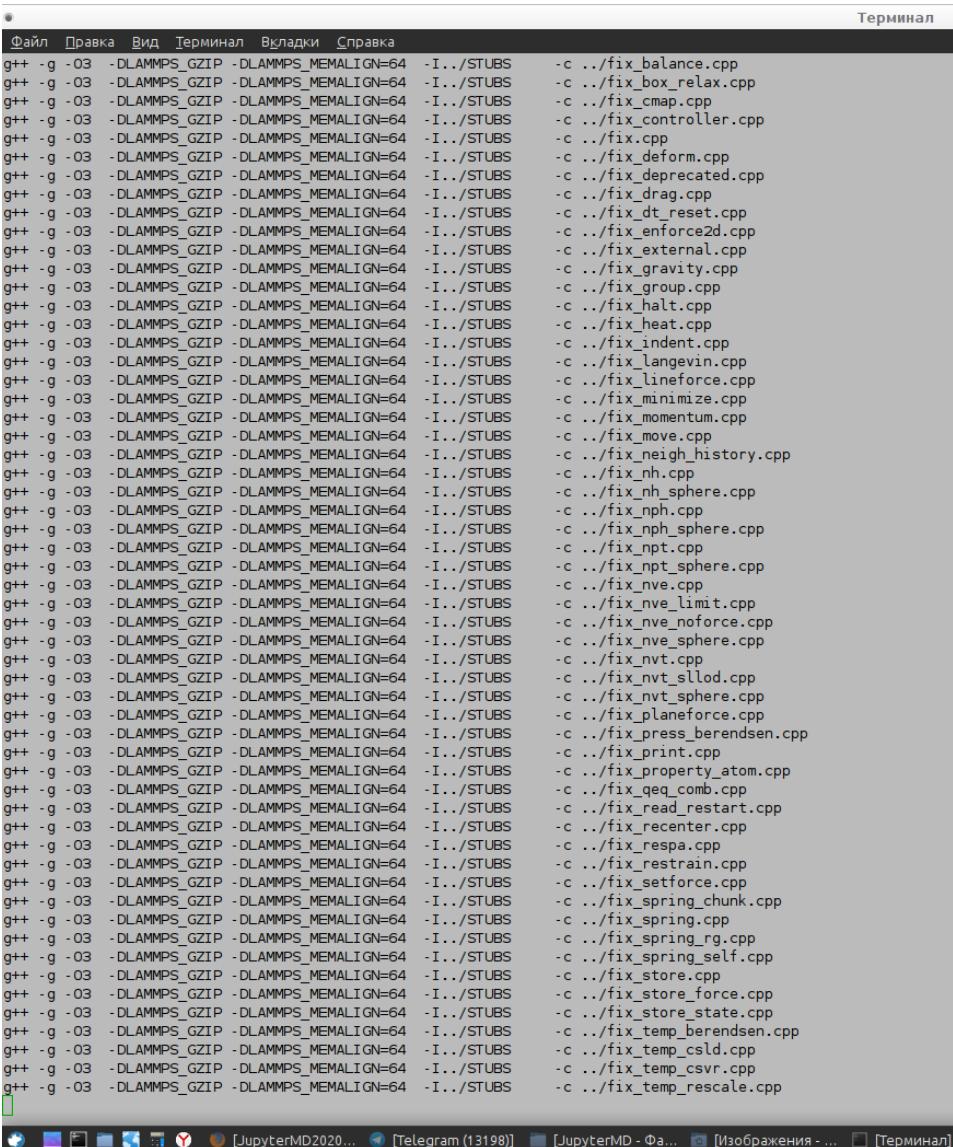
```
Файл Правка Вид Терминал Вкладки Справка
compute_dihedral.h           error.cpp
compute_dihedral_local.cpp   error.h
compute_dihedral_local.h     exceptions.h
compute_dipole_chunk.cpp    finish.cpp
compute_dipole_chunk.h      finish.h
compute_displace_atom.cpp   fix_adapt.cpp
compute_displace_atom.h     fix_adapt.h
compute_erootate_sphere_atom.cpp fix_addforce.cpp
compute_erootate_sphere_atom.h fix_addforce.h
compute_erootate_sphere.cpp  fix_ave_atom.cpp
compute_erootate_sphere.h    fix_ave_atom.h
compute_fragment_atom.cpp   fix_ave_chunk.cpp
compute_fragment_atom.h     fix_ave_chunk.h
compute_global_atom.cpp    fix_ave_correlate.cpp
compute_global_atom.h      fix_ave_correlate.h
compute_group_group.cpp   fix_aveforce.cpp
compute_group_group.h     fix_aveforce.h
compute_gyration_chunk.cpp fix_ave_histo.cpp
compute_gyration_chunk.h   fix_ave_histo.h
compute_gyration.cpp       fix_ave_histo_weight.cpp
compute_gyration.h         fix_ave_histo_weight.h
compute.h                   fix_ave_time.cpp
compute_heat_flux.cpp      fix_ave_time.h
compute_hexorder_atom.cpp  fix_balance.cpp
compute_hexorder_atom.h    fix_balance.h
compute_improper.cpp       fix_box_relax.cpp
compute_improper.h         fix_box_relax.h
compute_improper_local.cpp fix_controller.cpp
compute_improper_local.h   fix_controller.h
compute_inertia_chunk.cpp  fix.cpp
compute_inertia_chunk.h    fix_deform.cpp
compute_ke_atom.cpp        fix_deform.h
compute_ke_atom.h          fix_DEPRECATED.cpp
compute_ke.cpp              fix_DEPRECATED.h
compute_ke.h                fix_drag.cpp
compute_msd_chunk.cpp     fix_drag.h
compute_msd_chunk.h        fix_dt_reset.cpp
compute_msd.cpp             fix_dt_reset.h
compute_msd.h               fix_enforce2d.cpp
compute_omega_chunk.cpp   fix_enforce2d.h
compute_omega_chunk.h     fix_external.cpp
compute_orientorder_atom.cpp fix_EXTERNAL.h
compute_orientorder_atom.h fix_gravity.cpp
compute_pair.cpp            fix_gravity.h
compute_pair.h              fix_group.cpp
compute_pair_local.cpp    fix_group.h
compute_pair_local.h       fix_halt.cpp
(base) alex@alex-HP-ProBook-4730s:~/lammmps2020/src$ make clean-all
rm -rf Obj_*
(base) alex@alex-HP-ProBook-4730s:~/lammmps2020/src$ make yes-molecule
Installing package molecule
(base) alex@alex-HP-ProBook-4730s:~/lammmps2020/src$ make yes-kspace
Installing package kspace
(base) alex@alex-HP-ProBook-4730s:~/lammmps2020/src$ make yes-manybody
Installing package manybody
(base) alex@alex-HP-ProBook-4730s:~/lammmps2020/src$ make serial
```

Installation of LAMMPS

Case 3: Linux (via git)

1. Go to LAMMPS site
2. Go to Download page
3. Go to Git repository for LAMMPS
4. Copy string and past to terminal
5. Change branch: –b stable
6. Go to lammmps /src folder
7. make clean-all
8. Add packages: MOLECULE, KSPACE
9. Make LAMMPS serial version:

make serial



The terminal window shows a list of C++ files from the LAMMPS source code. The files are organized by their names and include various fix and compute commands. The terminal window has tabs at the bottom for JupyterMD2020..., Telegram (13198), JupyterMD - Файлы, Изображения - ..., and Терминал.

Файл	Правка	Вид	Терминал	Вкладки	Справка
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_balance.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_box_relax.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_cmap.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_controller.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_deform.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_DEPRECATED.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_drag.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_dt_reset.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_enforce2d.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_external.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_gravity.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_group.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_halt.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_heat.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_indent.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_langevin.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_lineforce.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_minimize.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_momentum.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_move.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_neigh_history.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nh.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nh_sphere.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nph.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nph_sphere.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_npt.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_npt_sphere.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nve.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nve_limit.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nve_noforce.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nve_sphere.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nvt.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nvt_sllod.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_nvt_sphere.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_planeforce.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_press_berendsen.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_print.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_property_atom.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_qeq_comb.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_read_restart.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_recenter.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_respa.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_restrain.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_setforce.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_spring_chunk.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_spring.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_spring_rg.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_spring_self.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_store.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_store_force.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_store_state.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_temp_berendsen.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_temp_csld.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_temp_csvr.cpp			
g++ -g -O3 -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64	-I.../STUBS	-c/fix_temp_rescale.cpp			

Installation of LAMMPS

Case 4: MS Windows

1. Go to LAMMPS Windows installer repository
2. Choose proper version

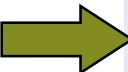
A screenshot of a web browser window. The address bar shows 'packages.lammps.org LAMMPS Windows Installer Repository'. Below the address bar is a blue navigation bar with a back arrow, a forward arrow, and a search icon. A link 'Back to LAMMPS Packages Overview' is visible. The main content area has a title 'LAMMPS Windows Installer Repository'. Below the title is a paragraph of text describing the repository's purpose and the packages it contains. To the right of the text is a small image showing a molecular simulation visualization and a Windows taskbar.

 **Some Notes on GPU Support**

These Windows binaries include GPU acceleration via the [GPU](#) package. This is achieved through compiling the GPU package in OpenCL mode and linking to an [OpenCL v1.2](#) compatible ICD loader. This means the executables do not contain any vendor provided code and should be compatible with GPUs from both [AMD](#) and [Nvidia](#). The GPU package has been compiled for mixed precision computation and is currently somewhat tuned for Nvidia (Kepler generation) GPUs. It [does not yet](#) work with OpenCL drivers for CPUs (like those included in the Intel and AMD OpenCL SDKs). In the case of having multiple Vendor provided OpenCL runtimes installed, you may run into the situation of the "wrong" runtime being set as the default and used by LAMMPS. In this case, you may need to remove unwanted vendors from the windows registry database. Look for the key: `HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors`, and remove vendors such as Intel. When reporting problems, please always include the exact [version](#) of the installer and the output of the `ocl_get_devices` tool.

Installing LAMMPS on Windows

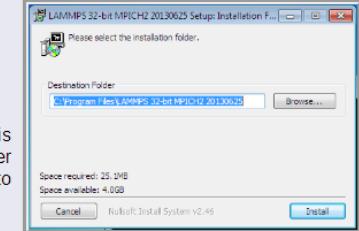
There are installer packages for 32-bit and 64-bit versions of Windows available.



- [Latest stable versions](#)
- [32-bit Windows download area](#) with all available installer versions
- [64-bit Windows download area](#) with all available installer versions

The respective download directory will contain installer packages that are labeled with the date of the LAMMPS version and packages labeled as *latest*. It is usually recommended to download and install the latest package. The other packages are provided in case there is a problem with it. Download the installer executable suitable for your machine, execute it, and follow the instructions in the dialogs. Each version will install into a different directory, so it is possible to have multiple versions installed at the same time (however it is not recommended). Both kinds of packages contain:

- Either a regular multi-threaded LAMMPS executable called `lmp_serial`. This should [always](#) work.
- Or a multi-threaded LAMMPS executable that also supports parallel execution via MPI message passing. This executable is called `lmp_mpi` and requires installation of a suitable MPICH2 package to work.
- the LAMMPS manual in PDF format
- the LAMMPS developer guide in PDF format
- the [colvars](#) reference manual in PDF format
- several additional PDF format guides for specific packages or styles
- the potential files bundled with the LAMMPS source code

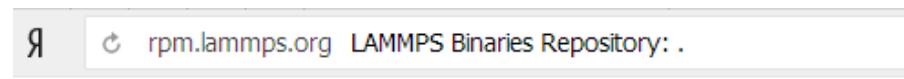
A screenshot of a Windows setup dialog box titled 'LAMMPS 32-bit MPICH2 20130625 Setup: Installation F...'. The dialog asks 'Please select the installation folder.' A 'Destination Folder' dropdown menu is open, showing 'C:\Program Files\LAMMPS 32-bit MPICH2 20130625'. Buttons for 'Cancel' and 'Install' are at the bottom. Below the dialog, text indicates 'Space required: 25 MB' and 'Space available: 4,000 MB'.

<http://packages.lammps.org/windows.html>

Installation of LAMMPS

Case 4: MS Windows

1. Go to LAMMPS Windows installer repository
2. Choose proper version (depends on your system)



LAMMPS Binaries Repository: .

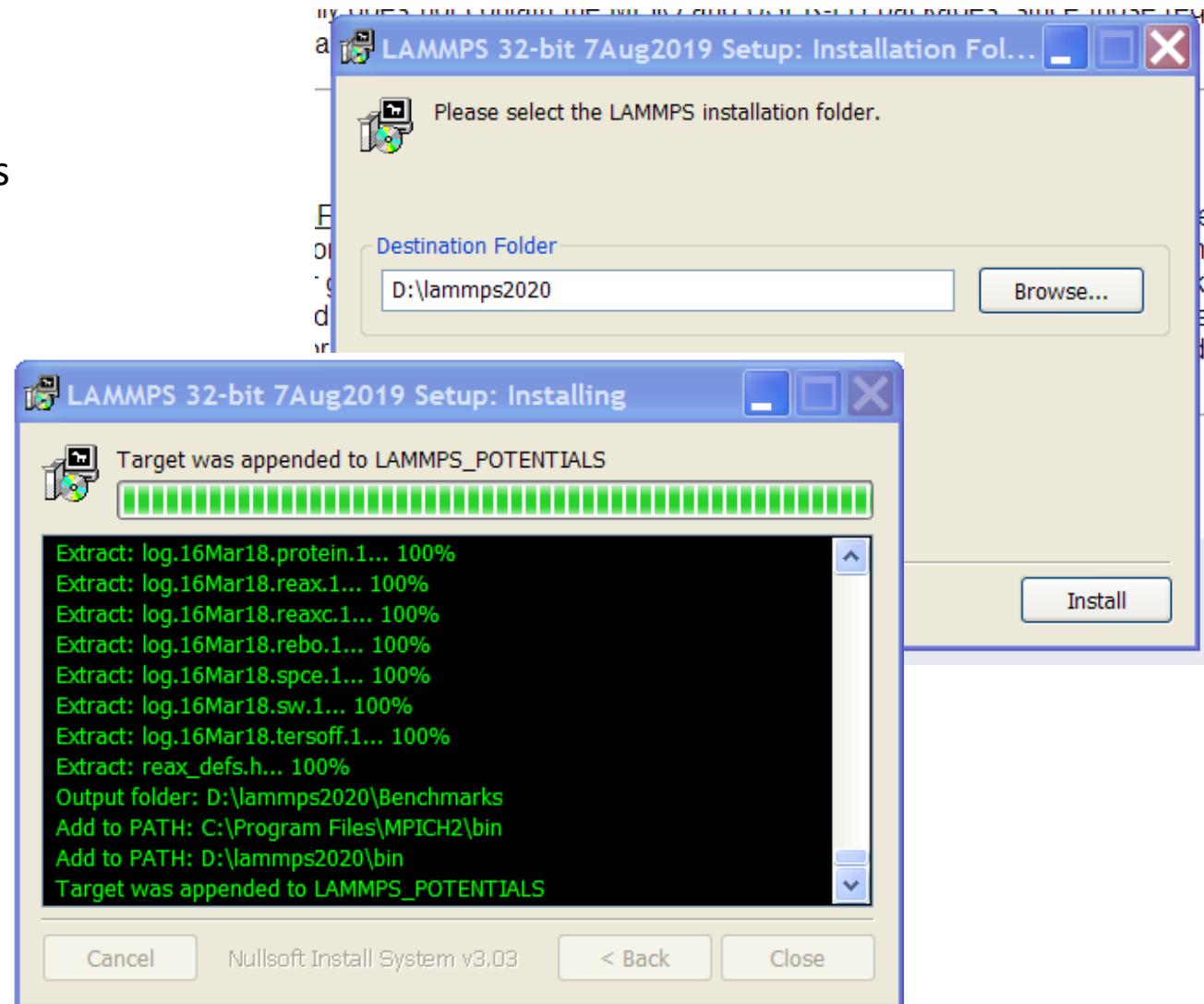
Contents of .

[DIR]	(Up one level)
[DIR]	32bit
[DIR]	64bit
[DIR]	snapshots
[DIR]	testing
2019-10-09 06:49	LAMMPS-64bit-7Aug2019.exe
2019-10-09 06:49	LAMMPS-64bit-7Aug2019-MPI.exe
2019-10-09 06:49	LAMMPS-32bit-7Aug2019.exe
2019-10-09 06:49	LAMMPS-32bit-7Aug2019-MPI.exe
2019-08-06 15:24	LAMMPS-64bit-stable-MPI.exe
2019-08-06 15:24	LAMMPS-32bit-stable-MPI.exe
2019-08-06 15:24	LAMMPS-64bit-stable.exe
2019-08-06 15:24	LAMMPS-32bit-stable.exe
2019-06-04 17:07	LAMMPS-64bit-5Jun2019.exe
2019-06-04 17:07	LAMMPS-64bit-5Jun2019-MPI.exe
2019-06-04 17:07	LAMMPS-32bit-5Jun2019.exe
2019-06-04 17:07	LAMMPS-32bit-5Jun2019-MPI.exe

Installation of LAMMPS

Case 4: MS Windows

1. Go to LAMMPS Windows installer repository
2. Choose proper version (depends on your system)
3. Download
4. Choose some folder
5. Run installation



Running of LAMMPS

or

```
lmp_serial -in <in-file> -e both
```

Input script with command for LAMMPS
– what LAMMPS must to do

```
lammmps -in <in-file> -e both
```

Option **-e** or **-echo** to output
log, warnings, errors etc.
(recommend to use “**both**”)

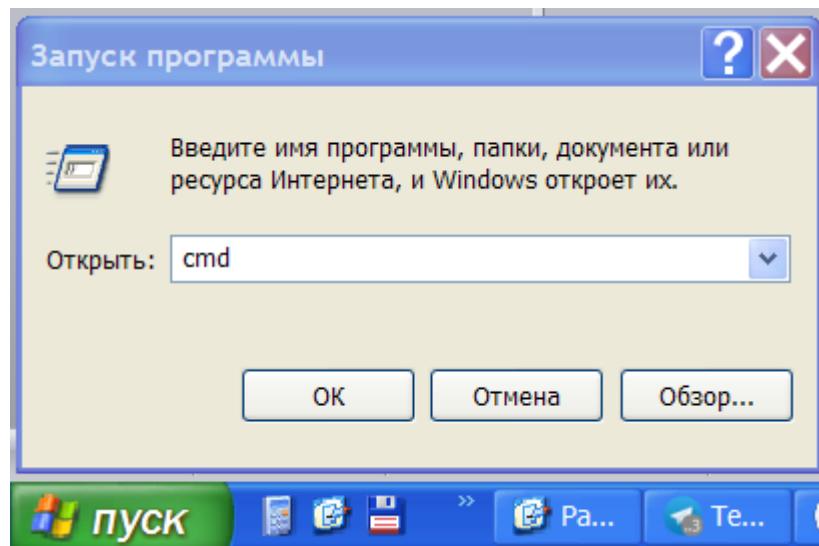
More about installation and running on different systems:

<https://lammps.sandia.gov/doc/Install.html>

https://lammps.sandia.gov/doc/Run_basics.html

Running of LAMMPS

Other way of running LAMMPS
in MS Windows: **Win+R** and **cmd**



The image shows a Windows Command Prompt window titled 'cmd C:\WINDOWS\system32\cmd.exe'. The window displays the output of a LAMMPS simulation. The output includes various neighbor list statistics and a summary of the run. At the bottom of the window, the command 'D:\lammmps2020\bin>lmp_serial.exe -in ../test/hw3.in -e both' is visible.

```
Pair      | 1.2031      | 1.2031      | 1.2031      | 0.0 | 0.76
Neigh    | 2.1562      | 2.1562      | 2.1562      | 0.0 | 1.36
Comm     | 6.1406      | 6.1406      | 6.1406      | 0.0 | 3.87
Output   | 0.20312     | 0.20312     | 0.20312     | 0.0 | 0.13
Modify   | 148.89       | 148.89       | 148.89       | 0.0 | 93.80
Other    |             | 0.1406       |             |     | 0.09

Nlocal: 64 ave 64 max 64 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Nghost: 714 ave 714 max 714 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Neighs: 2423 ave 2423 max 2423 min
Histogram: 1 0 0 0 0 0 0 0 0 0

Total # of neighbors = 2423
Ave neighs/atom = 37.8594
Neighbor list builds = 10000
Dangerous builds = 0

write_data gcmc.x.data
System init for write_data ...
Total wall time: 0:02:39

D:\lammmps2020\bin>lmp_serial.exe -in ../test/hw3.in -e both
```

More about installation and running on different systems:
<https://lammps.sandia.gov/doc/Install.html>
https://lammps.sandia.gov/doc/Run_basics.html

from lammps folder /bin/ or from everywhere:

lmp_serial < input-file

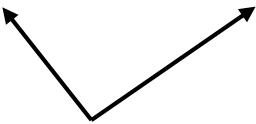
Additional software

- **VMD** – Visual Molecular Dynamics
 - <https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD>
- **Perl**
 - <https://www.perl.org/get.html>

Molecular **molecular dynamics** (MD)

(and HW3, tasks 1-4)

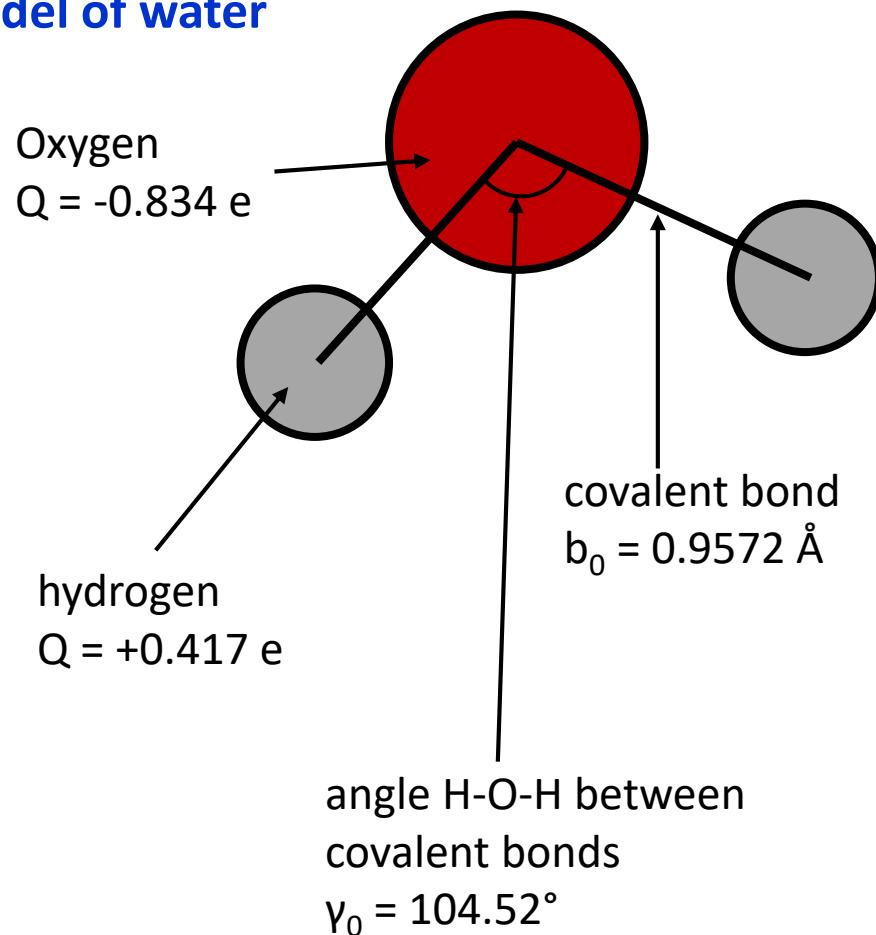
Molecular molecular dynamics



- Twice “molecular” is not a mistake!
- Previously we considered a simple LJ system, which is “atomic” (not “molecular”).
- LJ is a very accurate model, which describes the MD of inert (rare) gases
 - He, Ne, Ar, Kr...
- **HW3:** we will consider non-spherical molecules, namely, water, which is three-atomic compound – since we need to study the orientational correlation functions $K_u(t)$, $\Psi_{\theta,i}(t)$, that is an absurd in case of mono-atomic systems.

Molecular molecular dynamics

Model of water

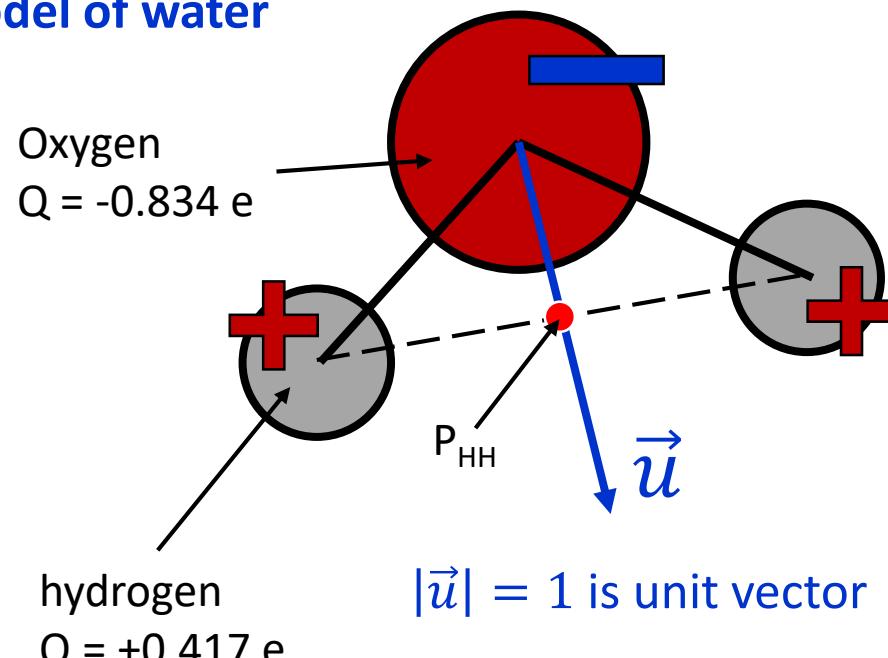


>> Three-site water model TIP3P (12 parameters):

- LJ parameters and mass of O
 - $m_O, \epsilon_O, \sigma_O$
- LJ parameters and mass H
 - $m_H, \epsilon_H, \sigma_H$
- Partial atomic charges for O and H (point electric charge approximation)
 - Q_O, Q_H
- Harmonic potential for covalent bonds (stiffness and equilibrium length of bond)
 - k_b, b_0
- Harmonic potential of angle H-O-H
 - k_γ, γ_0
- Bonded atoms (of the same water molecule) don't interact with each other (except via bond and angle).

Molecular molecular dynamics

Model of water



Orientation of water molecule:

>> Since oxygen atom is negatively charged and hydrogen atoms have positive charge,

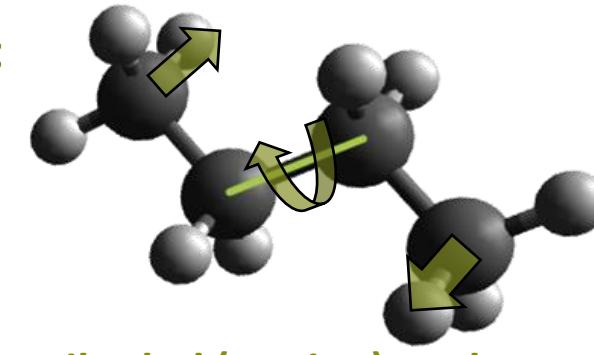
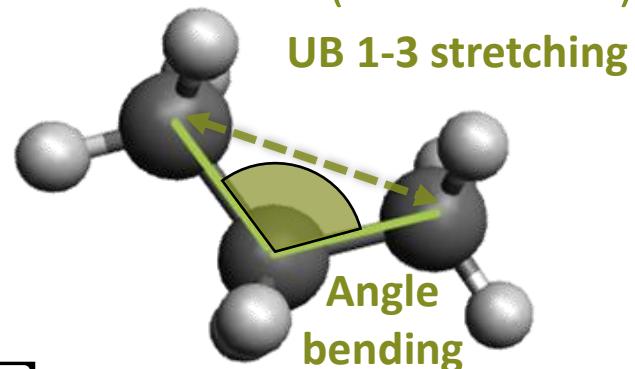
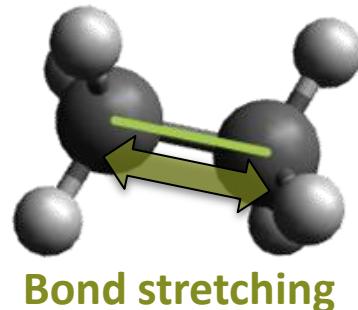
it is natural way to define orientation of water molecule as the unit vector \boldsymbol{u} directed from O atom through the median point P_{HH} between H atoms.

Note, the vector \boldsymbol{u} is colinear to electric dipole momentum of molecule.

$P_{\text{HH}} = \{x_p, y_p, z_p\}$ – is median point between H and H atoms

Force field

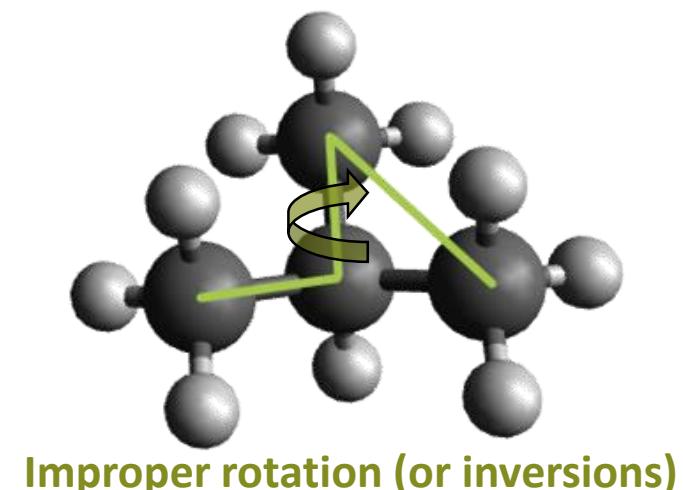
>> More general example of force fields (CHARMM FF) includes:



$$\begin{aligned}
 U = & \sum_{bonds} k_b(b - b_0)^2 + \sum_{angles} [k_\theta(\theta - \theta_0)^2 + k_{UB}(s - s_0)^2] + \\
 & + \sum_{dihedrals} k_\phi(1 + \cos(n\varphi - \delta))^2 + \sum_{impropers} k_\gamma(\gamma - \gamma_0)^2 + \\
 & + \sum_{i,j}^{n*n} 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} \right) + \sum_{i,j} \frac{q_i q_j}{\epsilon r_{ij}}
 \end{aligned}$$

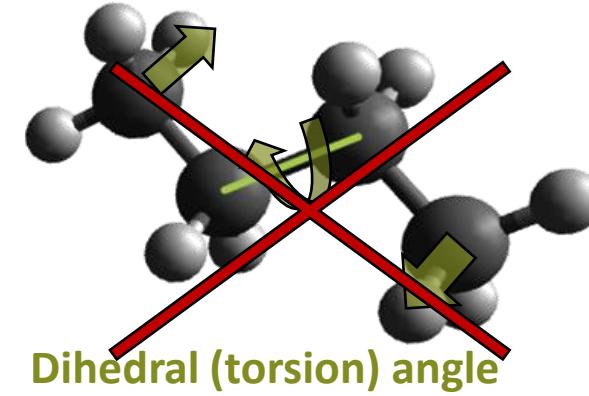
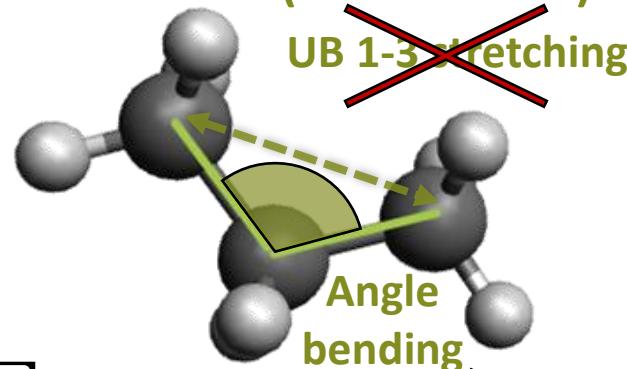
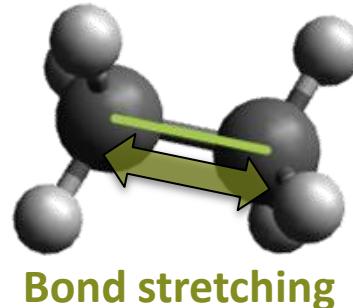
– typical form of potential energy functional of force fields (FF) like CHARMM

Lennard-Jones term **Coulombic term** **Unbonded terms**



Force field → water

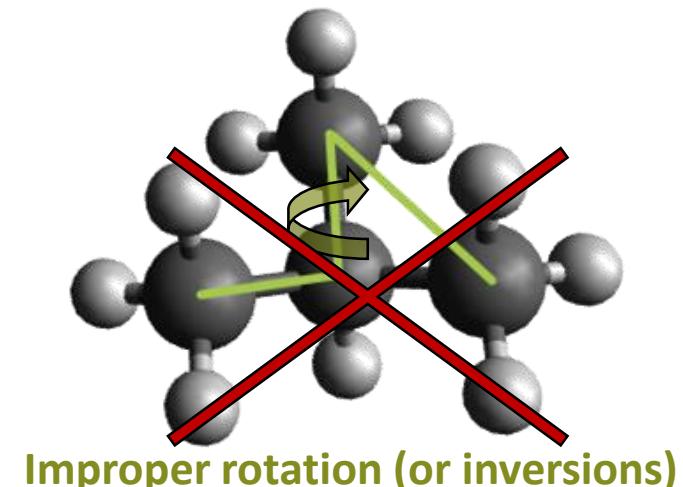
More general example of force fields (CHARMM FF) includes:



$$\begin{aligned}
 U = & \sum_{bonds} k_b(b - b_0)^2 + \sum_{angles} [k_\theta(\theta - \theta_0)^2 + k_{UB}(s - s_0)^2] + \\
 & + \sum_{dihedrals} k_\omega(1 + \cos(n\varphi - \delta))^2 + \sum_{impropers} k_\gamma(r - r_0)^2 + \\
 & + \sum_{i,j \neq j}^{n*n} 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} \right) + \sum_{i,j} \frac{q_i q_j}{\epsilon r_{ij}}
 \end{aligned}$$

– typical form of potential energy functional of force fields (FF) like CHARMM

Lennard-Jones term **Coulombic term** **Unbonded terms**



Force field: other examples

- **EAM, MEAM, ADP** – metals, metal alloys, metal nanoparticles...
- **REBO, Tersoff, Kolmogorov-Crespi** – reactive/“manybody” potentials...
- **ReaxFF** – allows study chemical reaction in C-O-H systems etc...
- **ANN FF** – artificial neural network-based potentials → **Final Project topic**

Embedded atom method (EAM):

$$E_{\text{total}} = \sum_i \left[F_{\alpha_i} \left(\sum_{j \neq i} \rho_{\alpha_j}(r_{ij}) \right) + \frac{1}{2} \sum_{j \neq i} V_{\alpha_i \alpha_j}(r_{ij}) \right]$$

Embedding energy term (of atom i)

Pairwise interaction energy term

Subscript α_i represents the type of atom with index i , ρ_{α_j} is the electron density function for atom type α_j

Electron density formed by j -atoms at the point of i -atom, which is considered

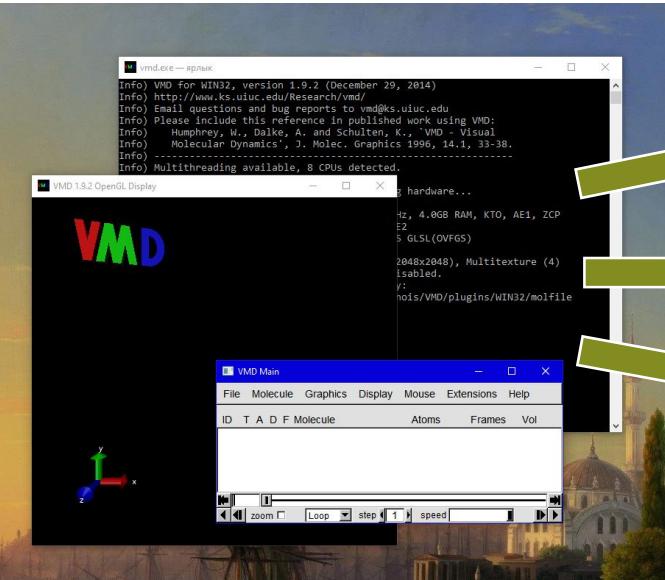
Interatomic distance

I recommend you to browse pages through: <https://lammps.sandia.gov/doc/pairs.html>
and this potential repository: <http://www.ctcms.nist.gov/potentials>

VMD

To use such complicated potentials as described above, we need some tools

Visual Molecular Dynamics (VMD)



Preparation of MD model

- Generation of structures
- Editing
- Merging
- Applying of the Force Field

Analysis of simulation results

Visual inspecting of model and visualization of the results

Download from: <https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD>

VMD. Functions. Examples.

- High-level tcl-based **Tk Console** (VMD Main → Extensions → Tk Console)
- Powerful **selection** logic
- Automatic topology generation (**psfgen** / **autopsf**)
- Useful modules (**topo**, **pbc** etc)

The screenshot shows the VMD TkConsole window with several command examples and associated annotations:

- Selection by residue (molecule) name**: Points to the command `(VMD) 1 % set s [atomselect top "resname TIP3"]`.
- Rotation of selected group of atoms**: Points to the command `>Main< (VMD) 2 % $s move [transaxis z 90]`.
- High-level selection logic**: Points to the command `>Main< (VMD) 3 % set sh [atomselect top "name H1 H2 and within 5 of not water"]`.
- Measuring of different quantities**: Points to the command `>Main< (VMD) 6 % measure center $i weight mass 0.4661850035190582 3.225283622741699 -1.556999921798706`.
- Wrapping coordinates to periodic box and preserve bonds**: Points to the command `>Main< (VMD) 7 % pbc wrap -all -compound resid -center origin`.
- Reading of positions and topology from LAMMPS data files ("full" and "atomic")**: Points to the command `>Main< (VMD) 8 % topo readlammpsdata 3H2O+NaCl.data full`.

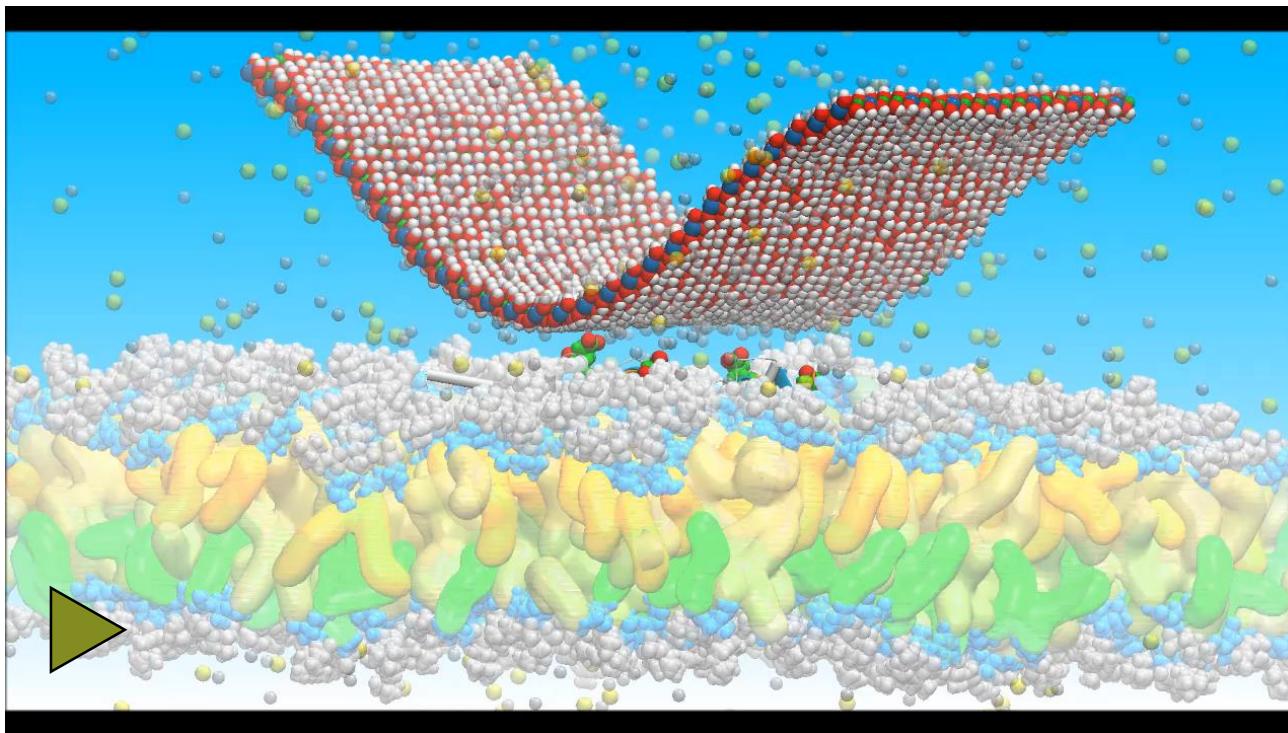
```
74 VMD TkConsole
File Console Edit Interp Prefs History Help
Main console display active (Tcl8.5.6 / Tk8.5.6)
(VMD) 1 % set s [atomselect top "resname TIP3"]
atomselect53
>Main< (VMD) 2 % $s move [transaxis z 90]
>Main< (VMD) 3 % set sh [atomselect top "name H1 H2 and within 5 of not water"]

atomselect54
>Main< (VMD) 4 % set $i [atomselect top "ions"]
can't read "i": no such variable
>Main< (VMD) 5 % set i [atomselect top "ions"]
atomselect55
>Main< (VMD) 6 % measure center $i weight mass
0.4661850035190582 3.225283622741699 -1.556999921798706
>Main< (VMD) 7 % pbc wrap -all -compound resid -center origin
Info) 100.0% complete (frame 0)
>Main< (VMD) 8 % topo readlammpsdata 3H2O+NaCl.data full
```

VMD. Example of complicated model

>> We will consider **CHARMM FF** due to several reasons as:

- It contains water models (including **TIP3P**), ions, almost all proteins, nucleotides, bio-membranes, nanomaterials **CHARMM-GUI** (<http://www.charmm-gui.org/?doc=input>)...
- It can be easily extended with FCC metals, polymers, silicon, ceramics by **INTERFACE FF**, **ClayFF_mod**
- and with drug molecules, as well as almost any small organic compounds by **SwissParam** (<http://swissparam.ch/>)



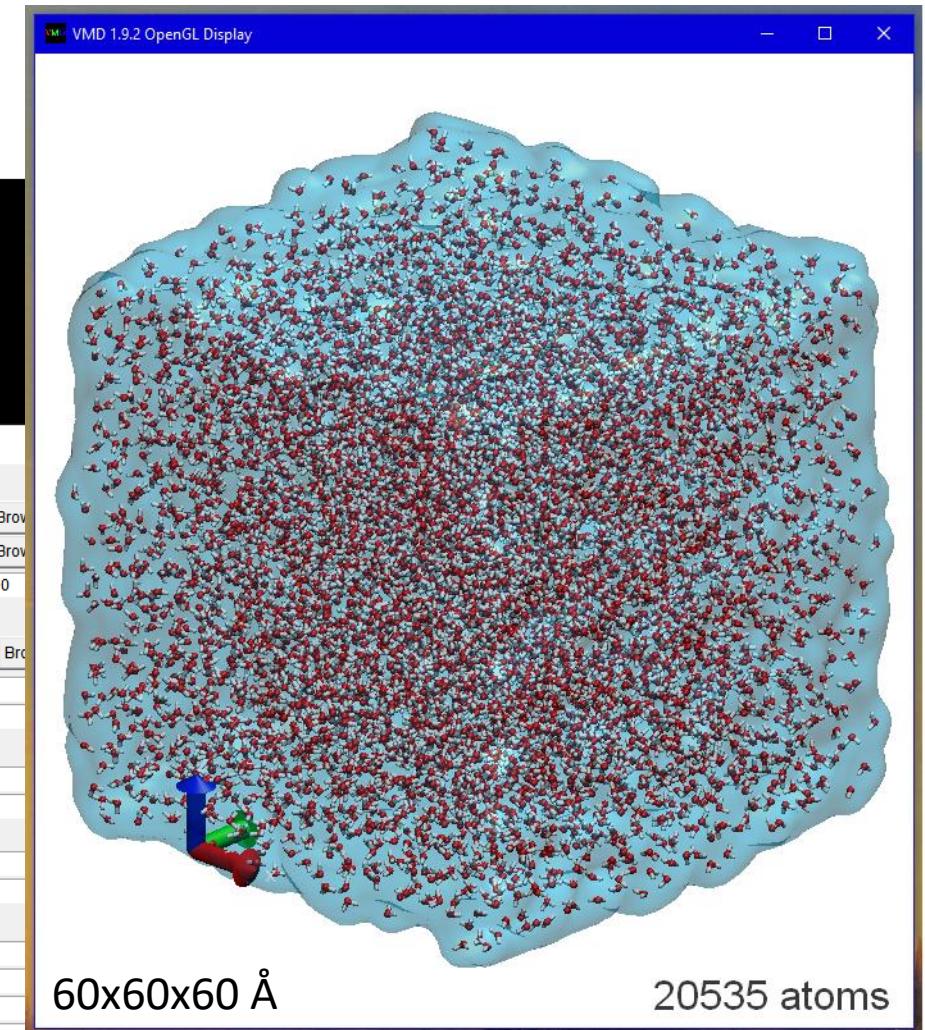
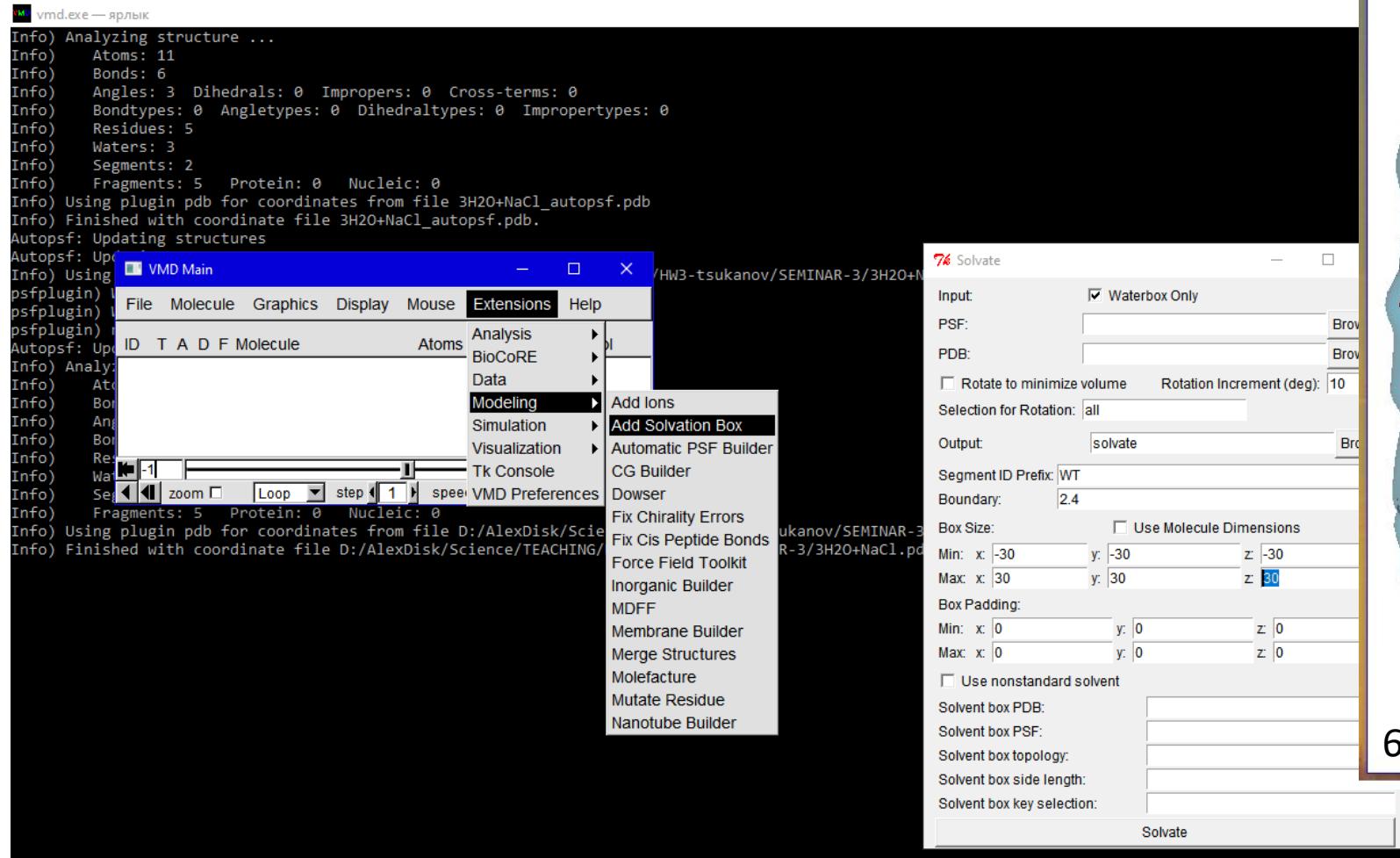
◀ Example of MD model (VMD,
CHARMM FF + ClayFF):

- Transmembrane protein (bacterial ion channel)
- Embedded into phospholipid bilayer
- In presence of Al hydroxide-based nanomaterial
- Whole system was put in the water with ions (water is not shown)

VMD. Preparation of TIP3P water model

Box filled with water can be easily obtained via

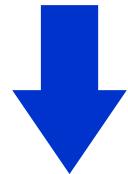
- VMD Main → Extensions → Modeling → Add Solvation Box



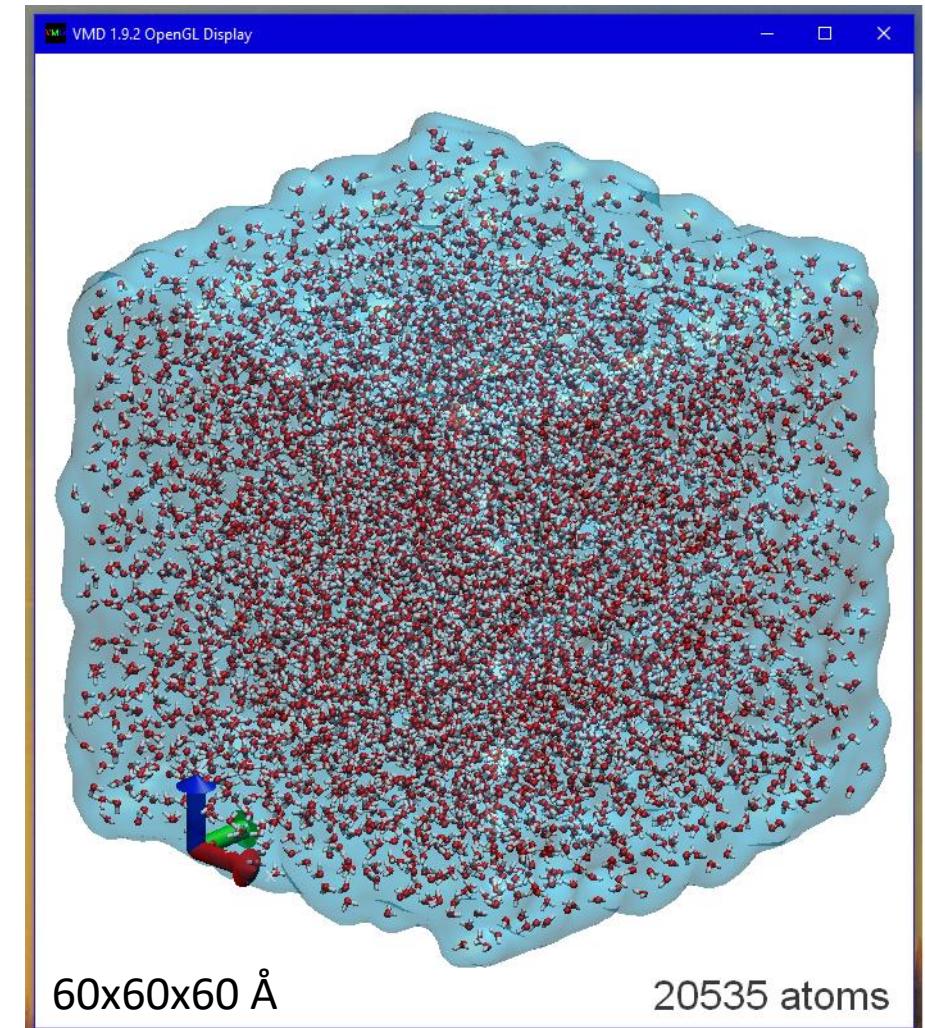
What is MD model?

>> **MD model** is

= SIMULATION BOX +
+ ATOMS (positions, names) + } **pdb file**
+ BONDS (topology) +
+ ANGLES + etc. + } **psf file**
+ types + charges +
+ **FORCE FIELD PARAMETERS**

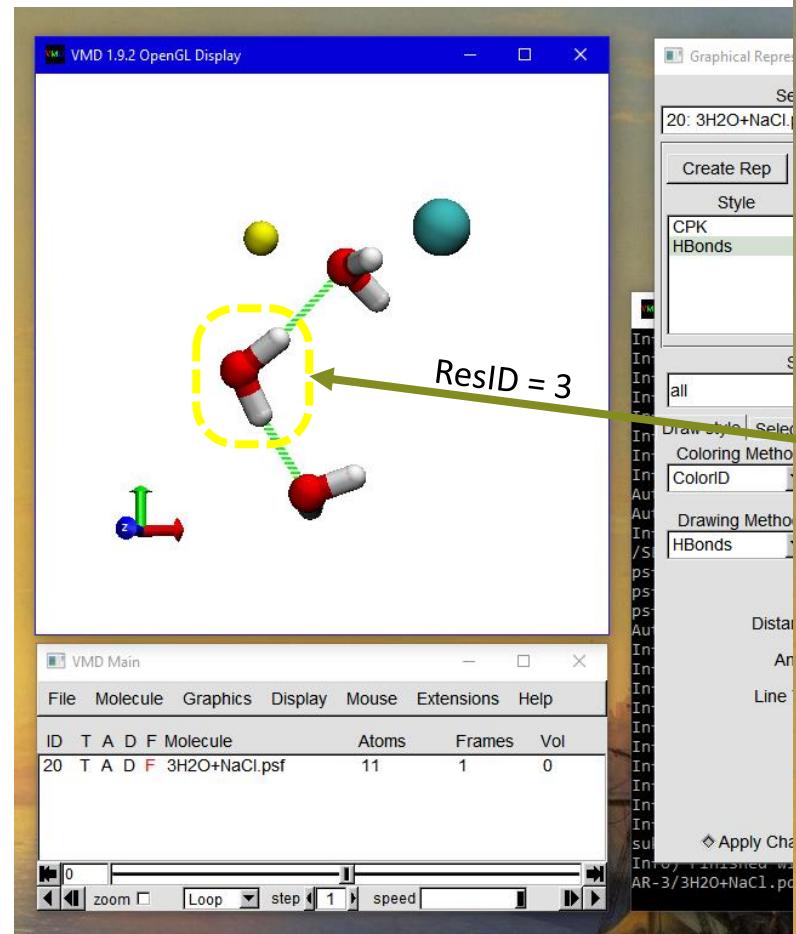


prm & **top** files
(charmm)



VMD. Example of PDB file

>> Structure of PDB file:



dimensions of the simulation box

D:\AlexDisk\Science\TEACHING\HW3-tsukanov\SEMINAR-3\3H2O+NaCl.pdb - Notepad++

Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ? X

3H2O+NaCl.pdb

	Index	Name	ResName	ResID	x, y, z in Å	SegName
1	CRYST1	7.000	7.000	7.000	90.00 90.00 90.00	P 1
2	ATOM	1 OH2	TIP3W	1	-0.180 -2.061 0.193	WT1 O
3	ATOM	2 H1	TIP3W	1	-0.482 -2.237 -0.705	WT1 H
4	ATOM	3 H2	TIP3W	1	0.718 -1.763 0.033	WT1 H
5	ATOM	4 OH2	TIP3W	2	1.154 2.217 1.523	WT1 O
6	ATOM	5 H1	TIP3W	2	1.559 1.626 0.895	WT1 H
7	ATOM	6 H2	TIP3W	2	1.909 2.285 2.134	WT1 H
8	ATOM	7 OH2	TIP3W	3	-1.079 0.275 1.558	WT1 O
	ATOM	8 H1	TIP3W	3	-0.816 -0.566 1.152	WT1 H
	ATOM	9 H2	TIP3W	3	-0.281 0.840 1.548	WT1 H
10	ATOM	10 SOD	SOD	1	-1.839 3.181 -1.557	ION NA
11	ATOM	11 CLA	CLA	1	1.961 3.254 -1.557	ION CL
12	END					
13						
14						

Nor length : 957

INS

Name must be unique within same ResName/ResID.
ResID must be unique within SegName.

VMD. TOPOLOGY (CHARMM FF)

pdb + top = psf 

C:\Program Files (x86)\University of Illinois\VMD\plugins\noarch\tcl\readcharmmtop1.1\top_all27_prot_lipid_na.inp - Notepad++

Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?

in-hw3-lj-fcc.tcl in-hw3-lj-liquid.tcl in-hw3lj-vacf-gas.tcl top_all36_lipid.rtf top_all27_prot_lipid_na.inp

```
1518 ic crp nr cr hr2 0.0 0.0 60.0 0.0 0.0
1519 ic hr nr cr hr3 0.0 0.0 120.0 0.0 0.0
1520 ic ca clp *nl hl 0.0 0.0 180.0 0.0 0.0
1521 ic ca nr *crp or 0.0 0.0 180.0 0.0 0.0
1522 ic hbl hb2 *cb hb3 0.0 0.0 120.0 0.0 0.0
1523 ic hll hl2 *cl hl3 0.0 0.0 240.0 0.0 0.0
1524 ic hrl hr2 *cr hr3 0.0 0.0 240.0 0.0 0.0
1525 ic ha ca nl hl 0.0 0.0 240.0 0.0 0.0
1526 patch first none last none
1527 RESI TIP3 total charge
1528
1529 { RESI TIP3 0.000 ! tip3p water model, generate using noangle nodihedral
1530 GROUP
1531 ATOM OH2 OT -0.834
1532 ATOM H1 HT 0.417
1533 ATOM H2 HT 0.417
1534 !BOND OH2 H1 OH2 H2 H1 H2 ! the last bond is needed for shake
1535 BOND OH2 H1 OH2 H2
1536 ANGLE H1 OH2 H2 ! required
1537 ACCEPTOR OH2
1538 PATCHING FIRS NONE LAST NONE
1539
1540 RESI TP3M 0.000 ! "mmff" water model, as an analog of tip3p
1541 GROUP
1542 ATOM OH2 OT -0.834 ! these charges are replaced by the mmff setup
1543 ATOM H1 HT 0.417 ! these charges are replaced by the mmff setup
1544 ATOM H2 HT 0.417 ! these charges are replaced by the mmff setup
1545 BOND OH2 H1 OH2 H2 ! omits the H1-H2 bond, which is needed for shake with tip3p
1546 ANGLE H1 OH2 H2 ! required
1547 ACCEPTOR OH2
1548 PATCHING FIRS NONE LAST NONE
1549
```

>> Here many “types”
(RESI) of molecules, ions,
residues are specified

<< description of compound
(single sample of molecule)

>> Here many “types”
(RESI) of molecules, ions,
residues are specified

<< description of compound (single sample of molecule)

```

1 PSF
2 !NTITLE
3 REMARKS segment WT1 3 molecules
4 REMARKS segment ION 2 ions
5
6      type   charge   mass
7
8      11 !NATOM
9      1 WT1    1     TIP3 OH2 | OT || -0.834000 | 15.9994 | 0
10     2 WT1    1     TIP3 H1  | HT || 0.417000 | 1.0080 | 0
11     3 WT1    1     TIP3 H2  | HT || 0.417000 | 1.0080 | 0
12     4 WT1    2     TIP3 OH2 | OT || -0.834000 | 15.9994 | 0
13     5 WT1    2     TIP3 H1  | HT || 0.417000 | 1.0080 | 0
14     6 WT1    2     TIP3 H2  | HT || 0.417000 | 1.0080 | 0
15     7 WT1    3     TIP3 OH2 | OT || -0.834000 | 15.9994 | 0
16     8 WT1    3     TIP3 H1  | HT || 0.417000 | 1.0080 | 0
17     9 WT1    3     TIP3 H2  | HT || 0.417000 | 1.0080 | 0
18
19
20      6 !NBOND: bonds << bonds for each molecule in system
21      1       2       1       3       4       5       6
22      7       8       7       9
23
24      3 !NTHETA: angles << angles for each molecule in system
25      2       1       3       5       4       6       8       7       9
26
27      0 !NPHI: dihedrals
28
29
30      0 !NIMPHI: impropers
31

```

VMD. Force field parameters (CHARMM FF)

prm
(par)
file>>

The screenshot shows a Notepad++ window displaying a CHARMM parameter file (par_wat.prm). The file contains sections for bonds, angles, and non-bonded interactions. A white box on the left side of the code area contains the text: "Parameters are specified only for atom types (not names or elements)". Three specific sections of the code are highlighted with red boxes and labeled with arrows:

- bond(s) parameters**: Located in the BONDS section, lines 10-12. It defines a bond between HT and OT atoms with parameters Kb = 450.0 and b0 = 0.9572.
- angle(s) parameters**: Located in the ANGLES section, lines 18-20. It defines an angle between HT, OT, and HT atoms with parameters Ktheta = 55.0 and Theta0 = 104.52.
- LJ parameters**: Located in the NONBONDED section, lines 29-30. It defines Lennard-Jones parameters for HT and OT atoms with epsilon = -0.046, Rmin/2 = 0.2245, and sigma = 1.7682.

Notable comments in the file include:
/* \\\\\\ CHARMM36 All-Hydrogen Lipid Parameter File ///////////////
/* All comments and questions should be submitted to the
/* parameter forum at the CHARMM website: www.charmm.org

```
* \\\\\\\ CHARMM36 All-Hydrogen Lipid Parameter File ///////////////
* All comments and questions should be submitted to the
* parameter forum at the CHARMM website: www.charmm.org
*
!reference in top_all36_lipid.rtf
BONDS
!V(bond) = Kb(b - b0)**2
!Kb: kcal/mole/A**2
!b0: A
!atom types Kb b0
HT OT 450.0 0.9572 ! from TIP3P geometry
ANGLES
!V(angle) = Ktheta(Theta - Theta0)**2
!Ktheta: kcal/mole/rad**2
!Theta0: degrees
!atom types Ktheta Theta0
HT OT HT 55.0 104.52 ! FROM TIP3P GEOMETRY
NONBONDED nbxmod 5 atom cdiel fshift vatom vdistance vfsswitch -
cutnb 14.0 ctfnb 12.0 ctonnb 10.0 eps 1.0 el4fac 1.0 wmin 1.5
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!epsilon: kcal/mole, Eps,i,j = sqrt(eps,i * eps,j)
!Rmin/2: A, Rmin,i,j = Rmin/2,i + Rmin/2,j
!atom ignored epsilon Rmin/2
HT 0.0 -0.046 0.2245
OT 0.0 -0.1521 1.7682
```

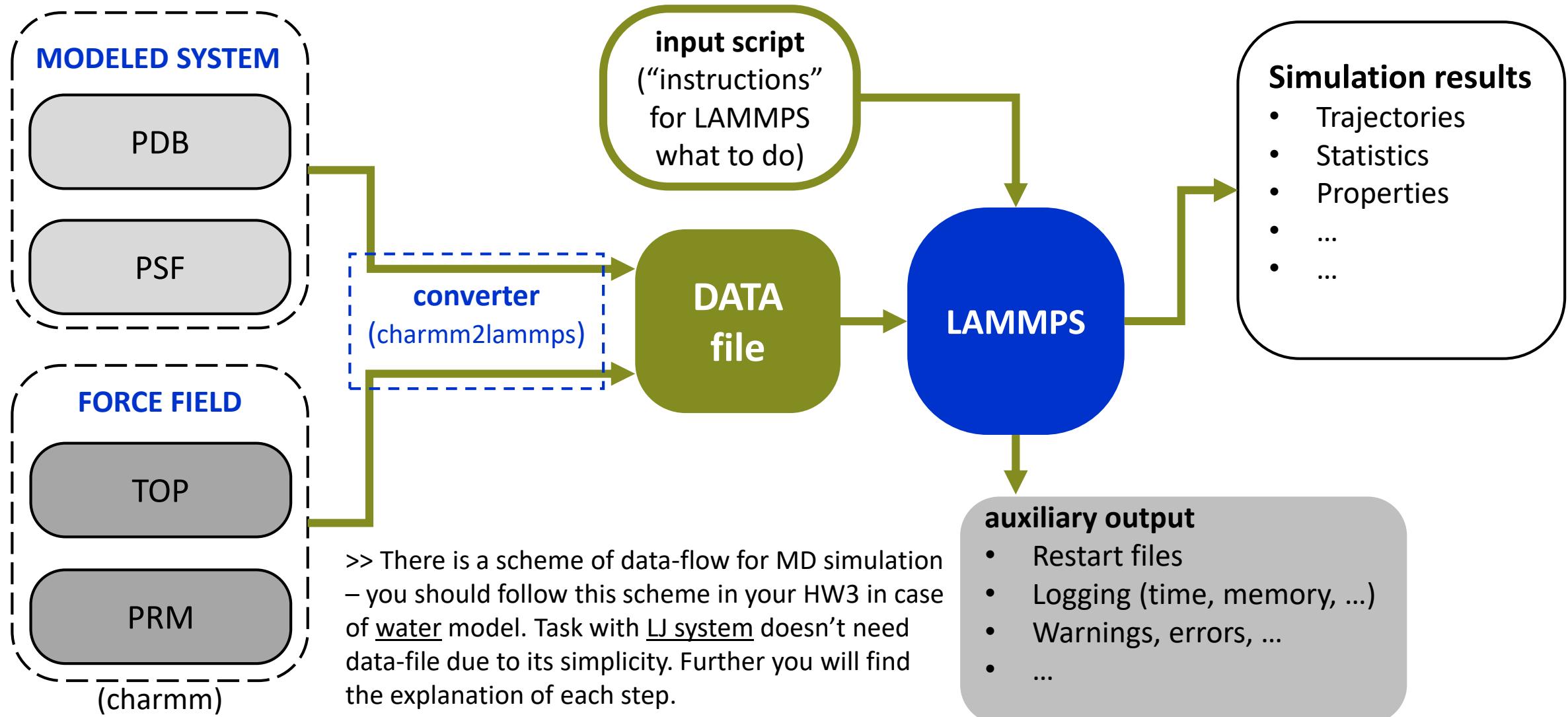
$$U_{bond}(i,j) = k_b(b - b_0)^2$$

$$U_{angle}(i,j,k) = k_\theta(\theta - \theta_0)^2$$

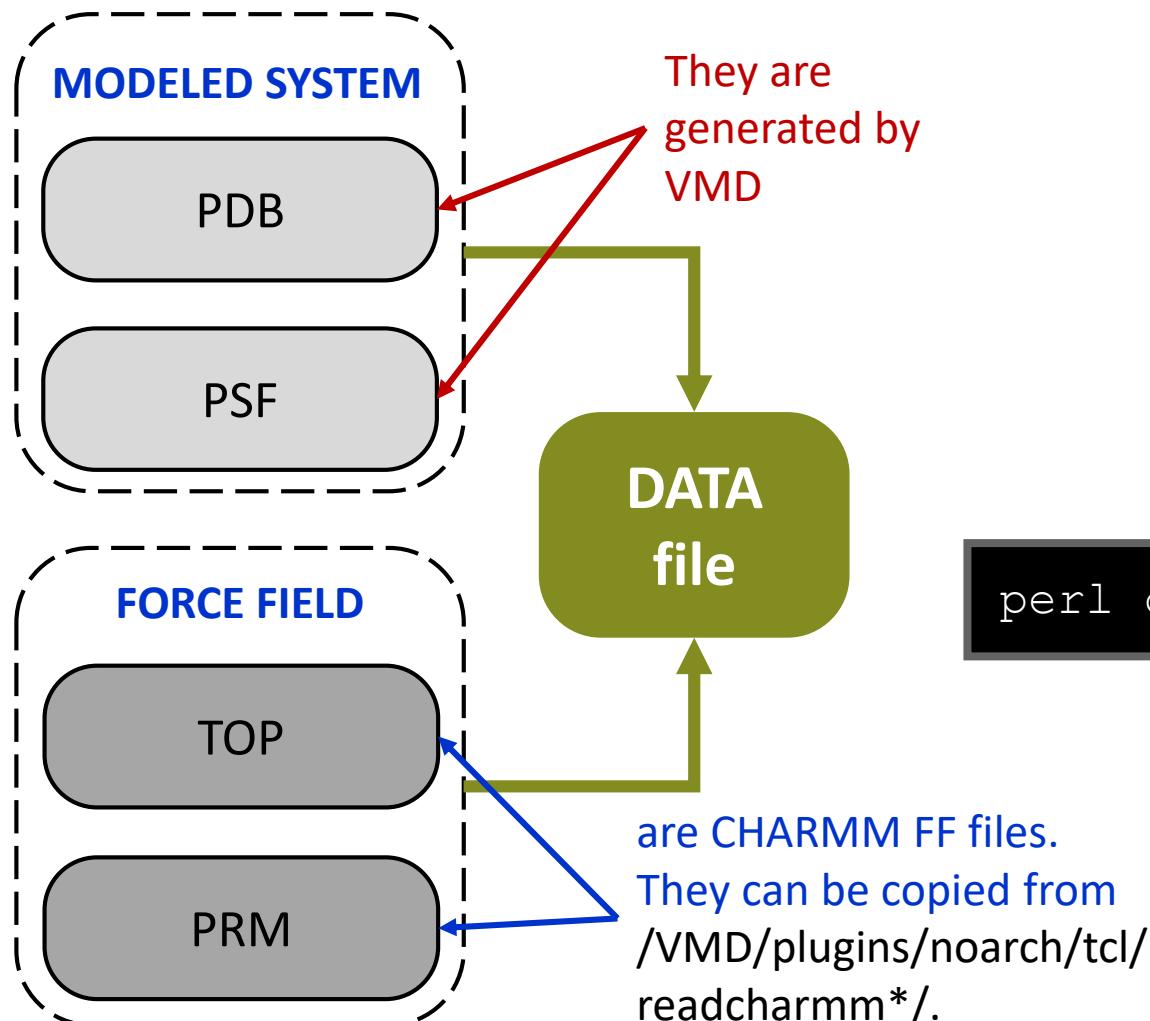
$$U_{LJ}: \epsilon = -\varepsilon$$

$$Rmin/2 = \frac{1}{2}\sigma\sqrt[6]{2}$$

Molecular dynamics ‘data-flow’



LAMMPS data-file



>> To built LAMMPS DATA-file 5 files should be gathered together in work directory:

1. *.pdb
2. *.psf
3. par_**.prm
4. top_**.rtf
5. charmm2lammmps.pl

Converter script can be copied from /LAMMPS/tools/

```
perl charmm2lammmps.pl -charmm -l=60 all36 solv
```

Example of command to convert **solv.psf** & **solv.pdb** to data-file **solv.data** using FF parameters and topology from **par_all36.prm** & **top_all36.rtf** files. Options **-lx**, **-ly**, **-lz** or **-l** specify dimensions of the computational domain. Option **-charmm** adds comments of atom types from CHARMM FF.

LAMMPS data-file

DATA file

Example of data-file for system of **233 TIP3P water molecules** in cube **20x20x20 Å**

```
D:\AlexDisk\Science\TEACHING\HW3-tsukanov\SEMINAR-3\solvate.data - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
solvate.data
1 Created by charmm2lammps v1.8.1 on Текущая дата: 27.01.2020
2
3 699 atoms
4 466 bonds
5 233 angles
6 0 dihedrals
7 0 impropers
8
9 2 atom types
10 1 bond types
11 1 angle types
12 0 dihedral types
13 0 improper types
14
15 -0.011      19.989 xlo xhi
16 -0.0065     19.9935 ylo yhi
17 0.08        20.08 zlo zhi
18
19 Masses
20
21 { 1      1.008 # HT
22   2     15.9994 # OT
23
24 Pair Coeffs
25
26 { 1      0.046  0.4000135    0.046  0.4000135 # HT
27   2     0.1521  3.150574    0.1521  3.150574 # OT
28
29 Atoms
Normal text file length : 1989 lines : 67 Ln : 1 Col : 61 Sel : 0 | 0 Windows (CR LF) OEM 866 INS .
```

System of N water molecules:

3N – atoms,
2N – bonds and
N – angles

besides that

2 types of atoms O and H,
1 type of bonds O-H,
1 type of angle H-O-H.

LAMMPS data-file

DATA file

Example of data-file for system of
233 TIP3P water molecules in
cube **20x20x20 Å**

List of atoms

```
D:\AlexDisk\Science\TEACHING\HW3-tsukanov\SEMINAR-3\solvate.data - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
X
solvate.data
18
19 Masses
20
21     1      1.008  # HT
22     2     15.9994 # OT
23
24 Pair Coeffs
25
26     1      0.046  0.4000135   0.046  0.4000135 # HT
27     2      0.1521  3.150574    0.1521  3.150574 # OT
28
29 Atoms id type charge x, y, z
30
31     1      1    -0.834   3.668  10.082  15.904 # OT
32     2      1     0.417   3.224  10.451  15.101 # HT
33     3      1     0.417   3.092  10.379  16.627 # HT
34     4      2    -0.834   5.186  16.696  12.072 # OT
35     5      2     0.417   6.083  16.979  11.816 # HT
36     6      2     0.417   5.337  15.832  12.49  # HT
37 ...
38     697    233    -0.834   3.347  10.487  19.814 # OT
39     698    233     0.417   2.792  10.955  19.188 # HT
40     699    233     0.417   4.264  10.639  19.456 # HT
41
42 Bond Coeffs
43
44     1      450     0.9572 # HT  OT
45
46 Bonds
```

Normal text file length : 1989 lines : 67 Ln : 1 Col : 61 Sel : 0 | 0 Windows (CR LF) OEM 866 INS

in comments:
charmm types
of atoms

LAMMPS data-file

DATA file

Example of data-file for system of **233 TIP3P water molecules** in cube **20x20x20 Å**

List of angle types

List of angles

```
D:\AlexDisk\Science\TEACHING\HW3-tsukanov\SEMINAR-3\solvate.data - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
X
solvate.data
38      697      233      2    -0.834      3.347      10.487      19.814 # OT
39      698      233      1     0.417      2.792      10.955      19.188 # HT
40      699      233      1     0.417      4.264      10.639      19.456 # HT
41
42 Bond Coeffs
43 bond type      kb      b0
44      1      450      0.9572 # HT      OT } List of bond types (only 1)
45
46 Bonds      #
47      1      1      1      2 # HT      OT
48      2      1      1      3 # HT      OT
49      3      1      4      5 # HT      OT
50      4      1      4      6 # HT      OT
51      ...
52      465      1      697      698 # HT      OT
53      466      1      697      699 # HT      OT } List of bonds
54
55 Angle Coeffs
56 Angle type      kθ      θ0
57      1      55      104.52      0      0 # HT      OT      HT
58
59 Angles      type of angle
60      1      1      2      1      3 # HT      OT      HT } List of angle types
61      2      1      1      4      6 # HT      OT      HT
62      ...
63      233      1      698      697      699 # HT      OT      HT } Triples of atoms in angle
64
65
66
```

Normal text file length : 1989 lines : 67 Ln : 1 Col : 61 Sel : 0 | 0 Windows (CR LF) OEM 866 INS

LAMMPS: Units

INPUT SCRIPT
(in-file)

UNITS – are important since all input and output values have units
LAMMPS supports many different systems of units (real, LJ, SI, metal...)
We will use **real** units

```
units real
```

Mass [m] = g/mol (grams per mole) = **amu**

Distance [r] = **Å** (angstrom) = 10^{-10} m

Time [t] = **fs** (femtosecond) = 10^{-15} s, **Velocity [v]** = **Å/fs**

Energy [U] = **kcal/mol**, **Force [F]** = **kcal/(mol Å)**

Temperature [T] = **K** (Kelvin)

Pressure [p] = **atm** (atmospheres) ~ 0.1 MPa

Charge [Q] = **e** (electrons)

Electric field [E] = **V/Å** (volts per angstrom)

Density [ρ] = **g/cm³**

For more details look at <https://lammps.sandia.gov/doc/units.html>

LAMMPS: PBC, (un)wrapped coordinates

INPUT SCRIPT (in-file)

>> To apply periodic boundary conditions along all axis
use command

```
boundary p p p
```

LAMMPS utilizes two “systems” of coordinates:

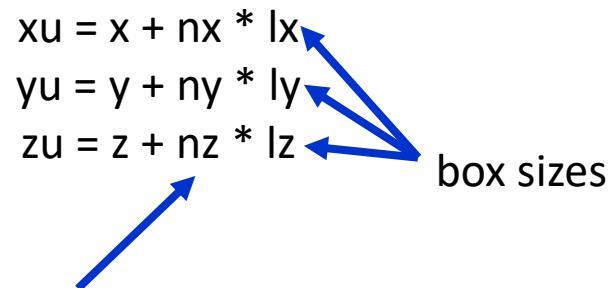
wrapped and unwrapped

x y z xu yu zu

They are related to each other by following relations:

$$\begin{aligned} xu &= x + nx * lx \\ yu &= y + ny * ly \\ zu &= z + nz * lz \end{aligned}$$

box sizes



In case of molecular systems we should save/dump coordinates in unwrapped form (for analysis and visualization) to prevent a breaking molecules in parts, and then using VMD we can wrap molecules in principal box taking into account bonds.

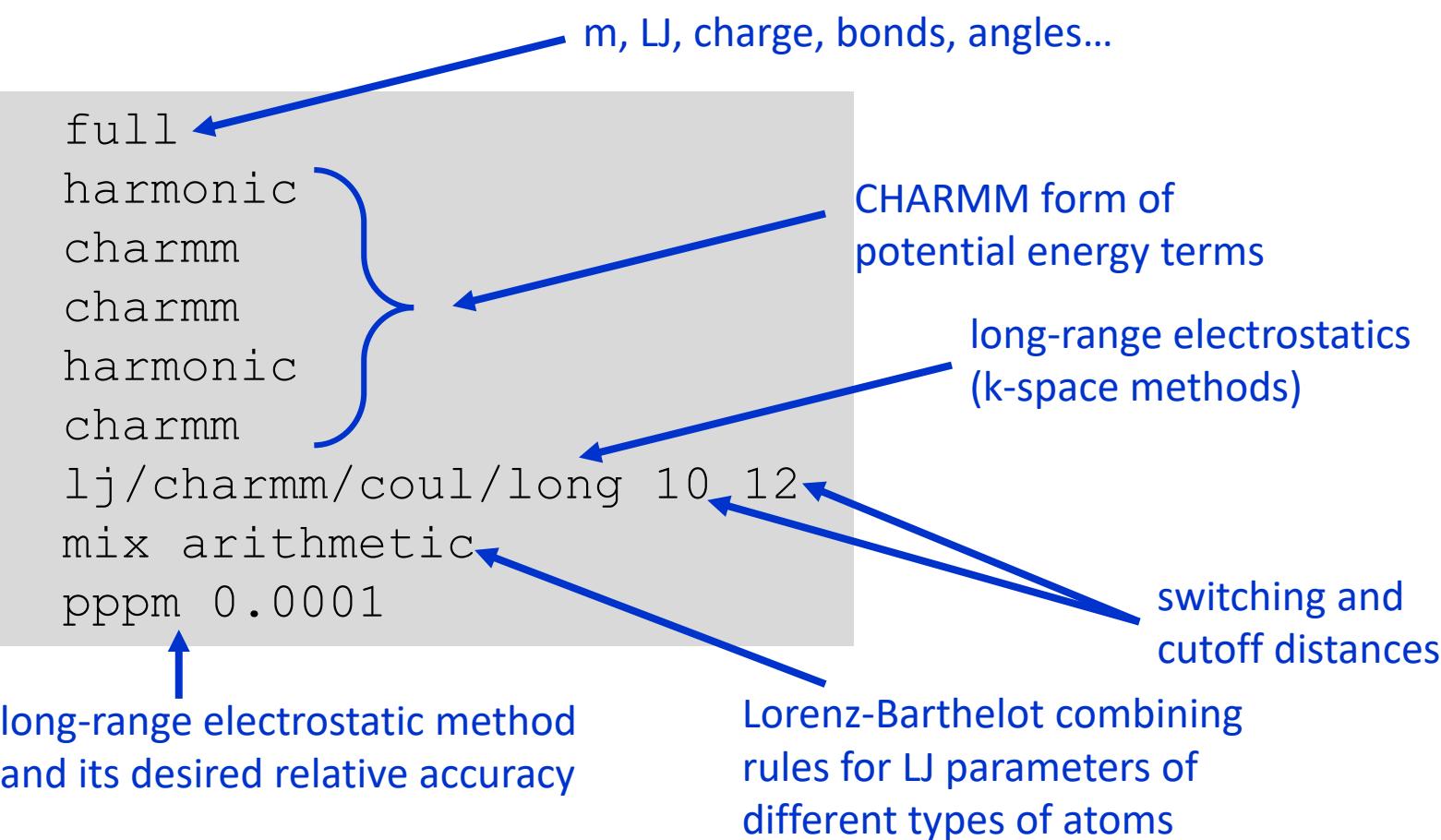
Images – how many times atom crossed the boundary (and in which direction), **nx**, **ny**, **nz** – can be negative, positive or zero

LAMMPS: styles of model

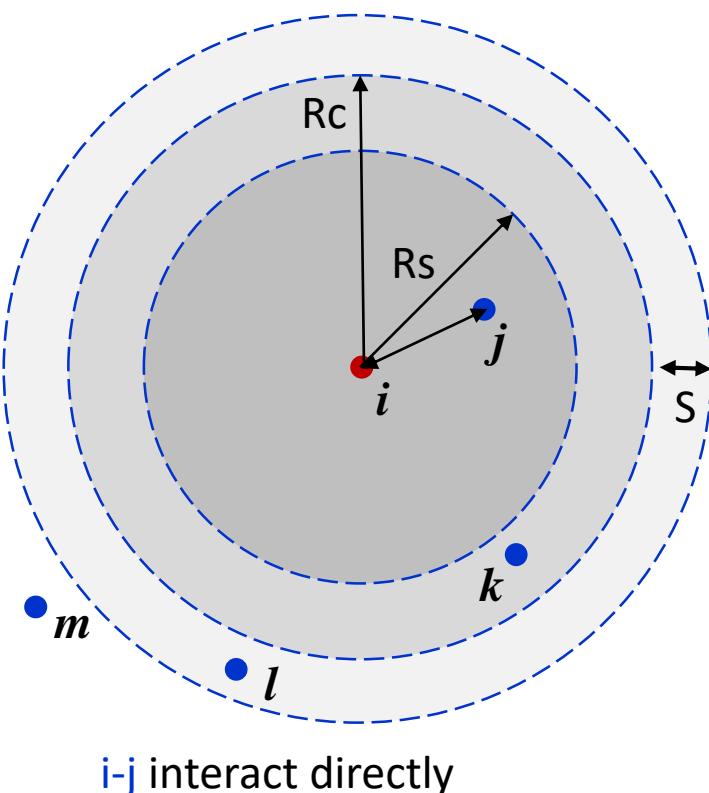
INPUT SCRIPT (in-file)

Style of atom description, pair interactions and potentials for bonded atoms for generated TIP3P model (as well as for CHARMM models) must be set as follows:

```
atom_style
bond_style
angle_style
dihedral_style
improper_style
special_bonds
pair_style
pair_modify
kspace_style
```



LAMMPS: cutoff distance, neighbor list



i-j interact directly

i-k interact with reduced force

i – in neighbor list of *i*, *m* – not.

>> As you remember, calculating pairwise interactions of atoms in HW1 we summed U_{ij} over all *i-j* pairs, that has complexity $\frac{1}{2}N^2$.

Due to the fact that LJ interaction at distances $>10 \text{ \AA}$ become quite small, LAMMPS calculates the pairwise interactions only among neighboring atoms. LAMMPS builds and updates neighbor list during simulation. Bonded atoms (1-2-3) are not included in neighbor list.

R_s – distance at which smoothly decaying envelop function is turned on (in previous case 10 \AA).

R_c – cutoff distance for LJ and electrostatics (in previous case 12 \AA) (long-range part of electrostatic interactions is treated by Ewald or PPPM methods).

S – so called skin – $R_c+S=R_{neigh}$ – radius for neighbor list (default $S=2 \text{ \AA}$)

It is recommended to use in your HW3 for updating neighbor list:

```
neigh_modify delay 1 every 1 check yes
```

LAMMPS input script: fixes

>> “FIXes” allow to control or to do something during simulation run:

```
fix 1 all nvt temp 300.0 300.0 100  
unfix 1
```

‘fix’ system at NVT conditions with target T = 300 K.
Hint: Use it in NVT equilibration.

```
fix 1 all npt temp 300.0 300.0 100 iso 1.0 1.0 1000
```

treats system at NpT conditions with desired T = 300 K and p = 1 atm
Hint: Use it in NpT equilibration and in main MD!

Examples of fixes for time-averaging (useful for averaging of RDF(r) function) and saving results into files

```
fix 2 all ave/time 20 10 200 c_1[*] ave running file RDF.txt mode vector  
fix 3 all print 100 "$(step) $(c_2[4])" file MSD.txt screen no
```

Hint: Use second example for output your MSD(t) function.

```
fix 2 all efield 0.0 0.0 0.05 ← This FIX sets electric field with Ez = 0.05 V/Å  
Hint: Use it in your HW3.
```

LAMMPS input script: groups

In many cases we need:

- to treat different parts of our system by different ways
- to calculate some properties for different parts of system
- to apply forces and constraints to some part of system etc.

LAMMPS allows to make groups

```
group frozen id 10:20           creates group of atom by atom id from 10 to 20
fix 1 frozen setforce 0.0 0.0 0.0   makes atoms of group "frozen" to be motionless
group mobile subtract all frozen    creates new group = all except "frozen",
velocity mobile create 300.0 20022020 mom yes rot yes dist gaussian      and set them random velocity
```

```
group hydrogens type 1
group oxygens type 2           creates group of atom by atom type
```

Hint: Make group by atom type to calculate MSD(t) only for oxygens in HW3.

LAMMPS input script: computes

To compute some quantity of the simulated system the “compute” commands can be used:

```
#md:          O-O  H-O
compute 1 all rdf 100 2 2 1 2
compute 2 oxygen msd com yes
```

Hint: Use this in your HW3.

comments line

computes RDF(r) for pairs 2-2 and 1-2

$0 < r \leq R_{\text{cutoff}}$, with resolution $R_{\text{cutoff}}/100$

Only atoms in neighbor list are taken into account!

Results are returned by “pointer” $c_1[...]$

computes MSD(t) of atoms in group “oxygen”,
motion of system center of mass is excluded.
Results are returned by “pointer” $c_2[...]$

```
compute 1 all vacf
```

Hint: Use this in your HW3.

computes VACF(t) for all atoms in system (group “all”).
Results are returned by “pointer” $c_1[...]$

LAMMPS input script: fix...print...file

To save the results of “computes” commands “fix...print...file” can be used:

```
compute 1 all vacf ← in previous case it computes VACF(t) and returns c_1[...] variable  
...  
fix 2 all print 10 "$(step) $(c_1[1]) $(c_1[4])" file vacf.txt title "s vacf_x  
VACF" screen no
```

Using variables `c_1[1], c_1[2], c_1[3] and c_1[4]` we can print results into file,
compute...vacf returns 4 values $Kv_x(t)$, $Kv_y(t)$, $Kv_z(t)$ and $Kv(t)$.

Hint: Use this to output VACF(t).

```
compute 2 oxygen msd com yes  
...  
fix 3 all print 100 "$(step) $(c_2[4])" file MSD.txt
```

Hint: Use this to output MSD(t).

LAMMPS input script: output with dump

Trajectories and the per-atom quantities can be saved into files via **damp** command:

```
dump 2 all dcd 100 frames.dcd  
#in case of molecular system coordinates should be unwrapped:  
dump_modify 2 unwrap yes
```

writes atom positions every 100 steps into DCD file, which can be visualized with VMD.

Per-atom values as coordinates, velocities, forces etc. can be written into files at every $N = 10$ steps by command:

```
dump 1 all custom 10 dump.unwrapped.*.txt mol type xu yu zu
```

Hint: Use this to save xu , yu , zu for all atoms into files and then read them via Python and calculate orientation of molecules and $K_u(t)$, $\Psi_{\theta_i}(t)$.

Hint: you can use this in all simulations for visual control of MD.

of molecule H or O coordinates

Instead of * lammps puts # of current timestep!
Thus, we'll obtain many files!

LAMMPS input script: thermo output

>> During simulations it is important to analyze behavior of such characteristics of system (or process) as energies, T, p, density etc.

To do this we can use “thermo” output by, for example, following way:

```
thermo_style custom step temp press lx ke pe etotal density  
thermo 100
```

and then we can use these values in fix...print command
(it is matter that Nevery = 100 in “fix...print” must be equal to *some N * Nevery* in “thermo”).

```
fix 3 all print 100 "$ (step) $(temp) $(press) $(lx:.3f)  
$(1000*density:.5g)" file thermo.txt title "Step  
T p L rho(kg/m3)" screen no
```

Hint: Use this in your HW3.

LAMMPS: example of input file

```
boundary      p p p
units         real
neigh_modify  delay 1 every 1 check yes

atom_style    full
bond_style    harmonic
angle_style   charmm
dihedral_style charmm
improper_style harmonic
special_bonds charmm
pair_style    lj/charmm/coul/long 10 12
pair_modify   mix arithmetic
kspace_style  pppm 0.0001

variable name index lesson3
#read data-file:
read_data solvate.data
#make folder for the results:
shell  mkdir ${name}.out
shell  cd ${name}.out
```

PBC, unit and neighbor list update options

Force field and model styles

...

Read data-file

Make directory for results (good style)

LAMMPS: example of input file

```
thermo_style      custom step temp press lx ke pe etotal density
thermo          100

#energy minimisation:
minimize 0.0 0.0 1000 10000

velocity all create 300.0 1234567 mom yes rot yes dist gaussian
#equilibration at NVT:
fix 1 all nvt temp 300.0 300.0 100
timestep 1.0
run 2000
unfix 1

#equilibration at NpT:
fix 1 all npt temp 300.0 300.0 100 iso 1.0 1.0 1000
run 4000
unfix 1
```

LAMMPS: example of input file

```
reset_timestep 0

#md:
compute 1 all rdf 100 2 2 1 2
fix 1 all npt temp 300.0 300.0 100 iso 1.0 1.0 1000
fix 2 all ave/time 100 40 4000 c_1[*] ave running file RDF.txt mode vector
fix 3 all print 100 "$(step) $(temp) $(density)" file T_and_rho.txt screen no

dump 1 all custom 1 dump.*.txt mol type x y z xu yu zu vx vy vz
dump_modify 1 sort id
dump 2 all dcd 100 ${name}.dcd
dump_modify 2 unwrap yes

run 4000
```

HW 3, task 1

1 Building a molecular dynamics model of water (2 points)

Download stable release of LAMMPS compatible with your PC (Sandia National Labs, USA, <https://lammps.sandia.gov/download.html>). Make (Linux) or install (MacOS, MS Windows) a serial version for desktop machines (*see Seminar/Lab 3 presentation*). Download and install VISUAL MOLECULAR DYNAMICS (VMD) software (<https://www.ks.uiuc.edu/>, version 1.9.2 is recommended¹). The programming language PERL should be installed, to convert input data using open script `charmm2lammps.pl`. The former can be found in `/LAMMPS/tools/` or via link

<https://github.com/lammps/lammps/blob/master/tools/ch2lmp/>.

Open terminal (in MS Windows, launch console via Win+R and command `cmd`), change working directory if it's necessary

`cd d:/path-to-dir-with-your-project/`

now you are able to run LAMMPS using the command

`lmp_serial -in _file_ -e both`

or

`lmp_serial < _file_ -e both`

or

`lammps -in _file_ -e both`, (depending on OS and way of installing) where `_file_` is your input script (*see Seminar/Lab 3 presentation for details*).

Hint. In Linux you can run LAMMPS directly from the directory, where it is installed `/path-to-lammps/src/lmp_serial -in....`

¹VMD of version 1.9.2 is recommended, since parameters of water model is included into common prm-files.

Build cubic computational domain $20 \times 20 \times 20$ Å, filled with water (model TIP3P). To do this use VMD program:

`VMD Main → Extensions → Modeling → Add Solvation Box`
Make sure that in input checkbox “Waterbox Only” is selected. You should get a model (pdb and psf files) containing about 700 atoms (230-240 water molecules), which is a suitable size of a model to simulate it in desktop PC (if you don't change output name, VMD generates files with name `solvate` by default).

Next step is to convert model into the lammps-data format using pdb, psf, topology (top-file) and parameters (par or prm) files as input and the Perl script `charmm2lammps.pl`.

The force field parameters file, for example `par_all36_lipid.prm`, can be copied from VMD subdirectory

`/VMD/plugins/noarch/tcl/readcharmpar#`

(where # is the number of version) to our work directory. Then rename it as `par_wat.prm`. Short topology file (`wat.top`) with only TIP3P water included can be found in VMD directory:

`/VMD/plugins/noarch/tcl/solvate#/`, copy it to our work directory and rename `top_wat.rtf`. Thus, to prepare lammps-compatible MD model we need to have at least 5 files gathered in our work directory:

- (1) `solvate.pdb`,
- (2) `solvate.psf`,
- (3) `top_wat.rtf`,
- (4) `par_wat.prm` and
- (5) `charmm2lammps.pl`.

Run the converter in terminal using the following command (make sure that current directory is your working one)

`perl charmm2lammps.pl -charmm -l=20 wat solvate`

The result should be `solvate.data` file, which describes modeled system in lammps-format. Make sure that all necessary blocks (atoms, bonds, angles) are written in the generated data-file.

HW 3, task 2

2 Modeling of the orientational motion of water molecules (6 points)

Using LAMMPS and TIP3P model of water from Task 1, calculate the orientational autocorrelation functions of water $K_u(t) = \langle \vec{u}(0) \cdot \vec{u}(t) \rangle$ and $\Psi_{\theta,l}(t)$ for $l = 2, 3$ (see the Lecture Notes, Lecture 5, page 9). Take the vector \vec{u} as a unit vector directed from the oxygen atom to the point between two hydrogen atoms of same molecule (see Seminar/Lab 3 presentation).

In all simulations use NpT ensemble² with $p = 1$ atm and $T = 300$ K, a recommended timestep δt is 1 fs. Save unwrapped³ coordinates of all atoms every 10-100 fs, where the whole simulation time should be at least 10000 fs (excluding equilibration). Then analyze the coordinates using Python (Jupyter Notebook). Include to your input LAMMPS script command `dump 2 ... dcd ...` to save trajectories. Don't forget to unwrap coordinates to prevent scission of bonds:

```
dump_modify 2 unwrap yes
```

(a) Plot the dependencies $K_u(t)$, $\Psi_{\theta,2}(t)$ and $\Psi_{\theta,3}(t)$. In order to control the system during the simulation as well as to be sure that the system is equilibrated, store T , p and L (the length of simulation box)⁴, energy (kinetic, potential, total) and density ρ of water to the file (you may use `fix ... print ... file` command) and then plot them versus time.

(b) Compute and compare the correlation times: τ_u , $\tau_{\theta,2}$ and $\tau_{\theta,3}$ (see L5 for the definitions).

(c) Apply the strong external electric field ($E_z = 0.05$ V/Å) to the system using the command `fix ... efield` and get an estimate of the orientation autocorrelation function in strong field $K_u^E(t)$. Plot the result and compare with field-free function $K_u(t)$. What is the difference? Is it correct to apply FDT in this case?

Open the initial psf-file (`solvate.psf`) in VMD and then load into it generated by LAMMPS dcd-file with computed trajectories:

VMD Main → File → Load Data Into Molecule...

Wrap coordinates using Tk Console of VMD (VMD Main → Extensions → Tk Console) and command

```
pbc wrap -all -compound resid -center origin
```

Are water molecules oriented along the electric field? Choose a representation style you like and make a screenshot of the studied system.

Hint 1. Don't forget first to equilibrate the system as NVT-ensemble and then at NpT conditions at least for $10^3 - 10^4$ fs. Don't reset the velocities after equilibration!

Hint 2. For saving the system thermodynamic parameters via

`fix ... print ... file` command it is necessary to add them in `thermo` output before. You can do this, for example, by:

```
thermo_style custom step temp press lx ke pe etotal density  
thermo 1000
```

Hint 3. It is convenient to save atomic coordinates at every 10 steps for the further analysis using `dump` command as follows:

```
dump 1 all custom 10 dump.*.txt mol type xu yu zu
```

Stored files can be easily read in Python via, for example, `scipy.genfromtxt()` function of SciPy module.

N, V and T are fixed (canonical ensemble) etc.

³see Seminar/Lab 3 presentation

⁴Since p is supposed to be fixed (in average), the volume, that is L^3 , may fluctuate, since it is not the control variable.

HW 3, task 3

3 Mean square displacement and pair distribution function (2 points)

Using the water model built in Task 1 and LAMMPS get an estimate of the diffusion coefficient D of water at $T = 300$ K and $p = 1$ atm and find the most probable distances between oxygen-oxygen d_{O-O} and oxygen-hydrogen d_{O-H} atoms of different molecules. To do this:

(a) Calculate the mean square displacement (MSD) averaged over oxygen atoms using command `compute ... msd` (don't take hydrogen atoms into account, make proper selection via `group` command). Save result into file with `fix ... print ... file` and then analyse in Python. The diffusion coefficient D can be estimated using the Einstein relation

$$6Dt = \lim_{t \rightarrow \infty} \langle |\vec{r}(t) - \vec{r}(0)|^2 \rangle \implies$$

$$D \approx \frac{1}{6\tau} \frac{1}{N} \sum_{i=1}^N |\vec{r}_i(\tau) - \vec{r}_i(0)|^2 = \frac{1}{6\tau} \text{MSD}(\tau)$$

where time τ must be at least $10^4 - 10^5$ fs. Plot $\text{MSD}(\tau)$. Do you know any other method to find D ? Which?

Hint. It is recommended to extract possible motions of the system center of mass (CoM), using the option `compute ... msd com yes`.

(b) Calculate the radial distribution functions (RDF) for O-O and O-H pairs using, for example, the command

```
compute 3 all rdf 100 2 2 1 2
```

where 2 and 1 are oxygen and hydrogen atom types, respectively. The result must be averaged and saved into a file via, for example, command `fix...ave/time... c_3[*] ave running file RDF.txt mode vector`⁵

Using Python, plot diagrams $\text{RDF}_{O-O}(r)$ and $\text{RDF}_{O-H}(r)$, find positions of the first local maximum d_{O-O} and d_{O-H} and compare the results with the typical donor-acceptor distance of hydrogen bond (H-bonds) in water, which is about 3 Å. Open `dcd`-file with the results in VMD, wrap coordinates, and add to representation the visualization of H-bonds: **VMD Main → Graphics → Representations → Drawing Method** choose HBonds. Make a screenshot of the studied system.

⁵Detailed description of the command can be found at https://lammps.sandia.gov/doc/fix_ave_time.html

HW 3, task 4

4 Solid, liquid and gaseous argon: VACF(t)* (2 points)

Calculate the velocity autocorrelation function VACF(t) for argon at different aggregate states. System must contain $10^3 - 10^4$ atoms. Use LAMMPS `units real`, `styles`:

```
atom_style atomic
pair_style lj/cut 8.5
pair_coeff * * 0.238 3.405
mass * 39.948
```

Lennard-Jones parameters and mass of argon can be specified by commands:

(a) To obtain the VACF of solid argon it is recommended to start simulation with face-centered cubic (FCC) crystal (see the explanations below, *Hint 3*) with a lattice constant of 5.26 Å. Try the temperature about 40 K and pressure about 100 MPa. Recommended duration of simulation is 2000-3000 fs. Plot VACF(t), does this function oscillate?

Hint 1. It is recommended to use `fix...box/relax iso...vmax 0.1` in energy minimization before the equilibration.

Hint 2. Note, the option `drag` may be useful for equilibrating your system with `fix...npt....`

Hint 3. You can generate the simulation system of 4000 Ar atoms with FCC crystal structure by:

```
lattice fcc 5.26
region box block 0 1 0 1 0 1 units lattice
create_box 1 box
create_atoms 1 box
replicate 10 10 10
```

(b) Liquid argon can be obtained by melting of FCC crystal (what parameters are to be changed to melt a crystal?). Recommended simulation time is 1500-2000 fs. Plot VACF(t).

Hint 4. Try, for example, T = 80 K and p = 0.1 MPa.

(c) Calculate VACF(t) and MSD(t) for gaseous Ar (how can you go from liquid to gas?). The simulation time should be at least $2 \cdot 10^6$ fs. Plot VACF(t) and compare with (a) and (b). Find an estimate for the diffusion coefficient D_{Ar} using the FDT relation

$$D = \frac{1}{3} \int_0^\infty \langle \vec{v}(0)\vec{v}(\tau) \rangle d\tau.$$

Compute the diffusion coefficient from the MSD (see the Problem 3). Compare two values.

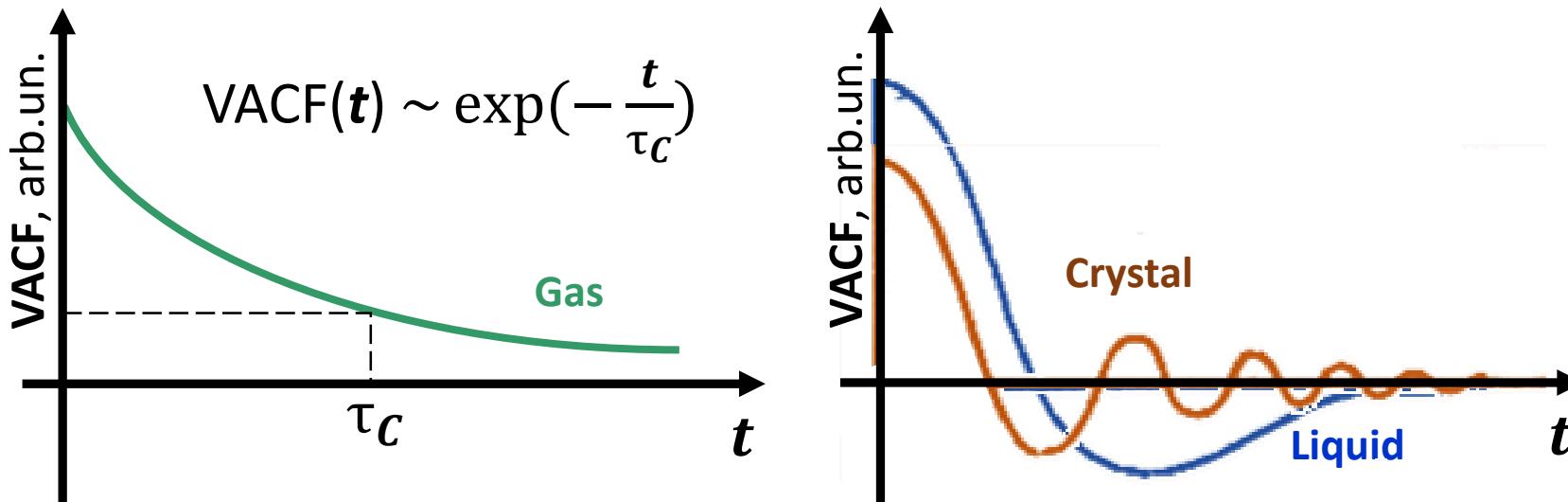
Thank you for your attention!

Some recommendations on MD, useful links:

- Book: Understanding Molecular Simulation (Academic Press) by **D. Frenkel** and **B. Smit**.
- Book: Computer Simulation of Liquids (Oxford Science Publications) by **M. P. Allen** and **D. J. Tildesley**.
- Link to **MDAnalysis** module for Python:
https://www.mdanalysis.org/mdanalysis/documentation_pages/overview.html
- Link to **OVITO** package homepage: <https://www.ovito.org/>
- Link to **Visual Molecular Dynamics** (VMD) download page:
<https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD>
- Open materials of **MIT** course “Atomistic computer modeling of materials” by **G. Ceder** and **N. Marzari**.

HW3: Velocity autocorrelation function of Ar

Behaviour of VACF (as a function of time) characterizes molecular thermal motion (*see Lecture notes*):



Velocity autocorrelation function depends on aggregate state of the matter.
=> In HW3 (task 4*): You should obtain all three cases.