

README

February 18, 2022

Software versions

- LAMMPS (29 Oct 2020)
- Python 3.6.9
- Matplotlib 3.3.4
- SciPy 1.5.4

Folder structure

Every script is meant to be run from the parent folder.

Dynamic brush

Main folder

Diffusion_bead_in_brush/

Initial configurations

Diffusion_bead_in_brush/Initial_configurations/SpacingX/Brush/Sigma_bead_1/

Hexagonal

Diffusion_bead_in_brush/Initial_configurations/SpacingX/Brush/Sigma_bead_1/Hexagonal/

Results

Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/Hexagonal/
Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/Nocut/
Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/Results/
Need to have indices_for_fit.txt or indices_for_fit_better_rms.txt in Nocut/ to get results.

Charged free beads

Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/ChargeY/
Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/ChargeY/Nocut/
Need to have indices_for_fit_Nestimates10_chargeY.tx in Nocut/ to get results.

Hexagonal

Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/Hexagonal/
Diffusion_bead_in_brush/SpacingX/damp10_diffseedLgv/Brush/Sigma_bead_1/Hexagonal/Nocut/
Need to have indices_for_fit.txt in Nocut/ to get results.

Collecting results for multiple d

Diffusion_bead_in_brush/D_vs_d/Bulk/Varysigmas/
Diffusion_bead_in_brush/D_vs_d/Brush/Sigma_bead_1/Nocut/
Diffusion_bead_in_brush/d_vs_Rg/
Old analysis folders that might still be linked:
Diffusion_bead_in_brush/D_vs_d/Bulk/d_vs_th/
Diffusion_bead_in_brush/D_vs_d/Bulk/d_vs_tr/

Static brush

Main folder

Diffusion_staticbrush/

Initial configurations

Diffusion_staticbrush/RadiusX/Initial_configs/Before_bead/
Diffusion_staticbrush/RadiusX/Initial_configs/

Results

Diffusion_staticbrush/RadiusX/
Diffusion_staticbrush/RadiusX/Results/
Diffusion_staticbrush/RadiusX/Nocut/
Need to have indices_for_fit.txt in Nocut/ to get results.

Collecting results for multiple d

Diffusion_staticbrush/D_vs_d/Nocut/

Scripts

Finding the persistence length:

`find_persistence_length_several_chains.lpstdv_corrected_2ndattempt.py`

Finding the persistence length, end-to-end distance, bond length, $\langle \cos \theta_1 \rangle$ and the z-coordinates of the last bead.

Bead-in-brush simulations

`makesystem_explicitsubstrate.py`

`in.brushdiffusion.lj_debye_angle_bonds.Langevin.ljunits_equilibrate_chains`

Create an equilibrated system of brushes on a substrate for insertion of diffusing bead.

`makesystem_placebead.py`

Takes the equilibrated system and places a bead at a random position near the substrate. At the same time, the script checks that the diffusing bead is not placed too close to the substrate or the beads in the chain. The script repeats this a number of times and writes the results to file.

`in.brushdiffusion.lj_debye_angle_bonds.Langevin.ljunits_readfromfile_varyradius`

Reads the data.-file generated using the two former scripts. This is where you set the radius of the diffusing bead. Only looped over the file number, i.e. you have to do run the script once for each combination of d and σ_b .

diffusion_bead_in_grid_manual_substrate_loop.py

Reads the MD data and finds $\langle R^2 \rangle$, $\langle dz^2 \rangle$ and $\langle dx^2 + dy^2 \rangle$. This is done in python and not in LAMMPS because we want to study the bead *in* the brush and not outside of it. Makes plots and writes to file. Also makes a feeble attempt to find D through a linear fit of all the data. Won't use these D 's and might remove that functionality. This script removes unphysical trajectories. (By 'unphysical trajectories' I mean trajectories where the bead bounces back and forth between the substrate and its periodic image)

Look at the plots

To determine where the graph is linear.

diffusion_replot_find_slopes_cutandnocut.py

To use this, you have to have run `diffusion_bead_in_grid` for both cut and no cut. Each direction is plotted separately.

Bead-in-brush simulations, no cuts

makesystem_explicitsubstrate.py

in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_equilibrate_chains

Create an equilibrated system of brushes on a substrate for insertion of diffusing bead.

makesystem_placebead.py

Takes the equilibrated system and places a bead at a random position near the substrate. At the same time, the script checks that the diffusing bead is not placed too close to the substrate or the beads in the chain. The script repeats this a number of times and writes the results to file.

in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_readfromfile_varyradius

Reads the data.-file generated using the two former scripts. This is where you set the radius of the diffusing bead. Currently only looped over the file number, i.e. you have to do run the script once for each combination of d and σ_b .

in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_readfromfile_varyradius_logfilesonly_potene

For finding the potential energy of the chains.

diffusion_bead_in_grid_dynamic_anomaliesaway.py

Reads the diffusion data from the script above and performs a fit. This extra step is included so that we can look at the plots and determine the suitable interval for the line fit used in finding D . Unfortunately, this must be done for every combination of d and σ_b . No way around it.

Look at the plots

To determine where the graph is linear.

diffusion_replot_find_slopes_cutandnocut.py

To use this, you have to have run `read_and_analyze` for both cut and no cut. Each direction is plotted separately. Make file `indices_for_fit.txt` and put in the `Nocut/` folder.

diffusion_avgandrms_dynamic_nocut_anomaliesaway.py

read_and_analyze_diffusion_nocut_better_rms.py

D vs d: gatherdata_diffusion_D_vs_d_nocut_better_rms.py

Gathers the diffusion coefficients vs d for a given σ .

radgyr_explicitsubstrate.py

Finding the radius of gyration of the chains in the bead-in-brush simulations (dynamic).

radgyr_writetogether.py

Gathering the data from `radgyr_explicitsubstrate.py`

plotavgenergy.py

Writing the potential energy of the free bead to terminal.

writeavgenergy_chain.py

Writing the potential energy of the chain bead to terminal.

Dd_dynamic_vs_static_nocut_Dpar_Dz_with_nostiffness_forest_better_rms.py

Generate plot D/D_{bulk} vs D . Needs to have a file for D_{bulk} first, and have run the gatherdata-scripts on all system types. This script also gives the exponential fit to D_{\perp}/D_{\parallel} .

plottogether_MDdynstat_and_RWfixedgeom_norefl_allmodels_fits.py

Performing fits and finding D vs a and diffusion times. Need to have run the gatherdata-scripts for dynamic and static systems (and bulk).

Charged brush

Use equilibrated files from dynamic brush

in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_readfromfile_varyradius_setcharge

diffusion_bead_in_grid_dynamic_anomaliesaway_setcharge.py

Nocut.

diffusion_bead_in_grid_manual_substrate_loop_setcharge.py

Cut.

diffusion_replot_find_slopes_cutandnocut_setcharge.py

Inspect plot and make `indices_for_fit_Nestimates10_chargeY.txt` to get results

diffusion_avgandrms_nocut_anomaliesaway_setcharge.py

gatherdata_diffusion_D_vs_d_nocut_setcharge.py

plottogether_D_vs_d_charges.py

R_vs_d.py

Finding the conductivity σ_e , resistivity ρ_e and the resistance R .

Bead-in-brush simulations, hexagonal grid

`in.set_hexagonal_tethergrid`

`make_hexagonal_withsubstrate.py`

`in.brushdiffusion_HEX_lj_debye_angle_bonds_Langevin_ljunits_equilibrate_chains`

`makesystem_placebead_hexagonal.py`

`in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_readfromfile_varyradius_hexagonal`

`diffusion_bead_in_grid_dynamic_hexagonal.py`

`diffusion_bead_in_grid_dynamic_anomaliesaway_hexagonal.py`

`diffusion_replot_find_slopes_cutandnocut_hexagonal.py`

`read_and_analyze_diffusion_nocut_hexagonal.py`

`gatherdata_diffusion_quadr_vs_hex_nocut.py`

Bead-in-brush simulations, no stiffness

`makesystem_explicitsubstrate.py`

`in.brushdiffusion_equilibrate_nostiffness`

`makesystem_placebead_nostiffness.py`

`in.brushdiffusion_nostiffness`

`diffusion_bead_in_grid_nostiffness_anomaliesaway.py`

`diffusion_bead_in_grid_nostiffness_cut.py`

Will run this too, in order to find a suitable interval.

`diffusion_replot_find_slopes_cutandnocut_nostiffness.py`

`diffusion_avgandrms_nostiffness_nocut_anomaliesaway.py`

`gatherdata_diffusion_D_vs_d_nostiffness_nocut_better_rms.py`

Bead-in-static-brush simulations

`makesystem_explicitsubstrate.py`

`in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_makestaticconfigs`

`makesystem_placebead_to_static.py`

`makesystem_placebead_to_static_varyradius`

This is the script I prefer now. Have a few cases where I tried to use different radii, and the naming kind of got stuck after that.

`in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_staticbrush`

`in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_staticbrush_different_sigma_b`

`diffusion_bead_in_grid_static.py`

All unphysical trajectories are discarded.

`diffusion_replot_find_slopes_cutandnocut_static.py`

To use this, you have to have run `read_and_analyze` for both cut and no cut. Each direction is plotted separately. Write `indices_for_fit.txt`

Bead-in-static-brush simulations, no cut

`makesystem_explicitsubstrate.py`

`in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_makestaticconfigs`

`makesystem_placebead_to_static_varyradius`

This USED to be the script I preferred. Have a few cases where I tried to use different radii, and the naming kind of got stuck after that.

`makesystem_placebead_to_static_indepconfigs_varyradius.py`

This is the script I prefer now. I need more statistics, so I need more brush configs.

`in.brushdiffusion_lj_debye_angle_bonds_Langevin_ljunits_staticbrush_different_sigma_b`

`diffusion_bead_in_grid_static_anomaliesaway.py`

`diffusion_replot_find_slopes_cutandnocut_static.py`

To use this, you have to have run `read_and_analyze` for both cut and no cut. Each direction is plotted separately. After inspecting the graph, make `indices_for_fit.txt` in `Nocut/`

`diffusion_avgandrms_static_nocut_anomaliesaway.py`

D vs d : `gatherdata_diffusion_D_vs_d_static_nocut_better_rms.py`

Gathers the diffusion coefficients vs d for a given σ . Reads results from `read_and_analyze_diffusion_static.py`, which in turn treats the results from `read_and_analyze_diffusion_static.py` or `diffusion_bulk.py`.

`read_and_analyze_diffusion_static_nocut_better_rms.py`

To better estimate uncertainty.

D vs d: gatherdata_diffusion_D_vs_d_static_nocut.py

D vs d: gatherdata_diffusion_D_vs_d_static_nocut_better_rms.py

Bead in straight system, cut

`makesystem_placebead_forest.py`

`in.forest_lj_debye_Langevin_units`

`diffusion_forest_cut.py`

`diffusion_replot_find_slopes_cutandnocut_forest_nocut.py`

Bead in straight system, no cut

`makesystem_placebead_forest.py`

`in.forest_lj_debye_Langevin_units`

`diffusion_forest.py`

`diffusion_replot_find_slopes_forest_nocut.py`

`read_and_analyze_diffusion_forest_nocut.py`

`gatherdata_diffusion_D_vs_d_forest_nocut.py`

Bulk diffusion

`makesystem_emptybox.py`

`in.bulkdiffusion_withoutsubstrate`

`diffusion_bulk_pure.py`

Perform analysis on bulk simulations (without substrate).

`diffusion_avgandrms_bulk_anomaliesaway.py`

Find values with better rms values.

`gatherdata_diffusion_D_vs_sigma_bulk_better_rms.py`

Collect and plot D vs sigma.

Notes

There are slightly different setups for the `indices_for_fits`-files. If you want to recreate the results using the scripts, look at the `diffusion_avgandrms`-scripts in order to find the setup.