Worksheet

Tutorial: Under the hood

Axel Arnold

October 8, 2012

1 Getting ESPResSo

Use the command

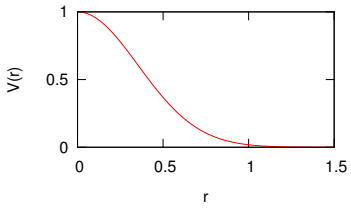
```
git clone git://git.savannah.nongnu.org/espressomd.git
```

to get the latest ESPResSo code.

2 Adding a Gaussian potential

So far ESPResSo does not provide a Gaussian potential, which has a finite overlap energy with a plateau value. We will now add this potential, using the form:

$$V(r) = \begin{cases} \epsilon\, e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2} & r < r_{\mathrm{cut}} \\ 0 & r \geq r_{\mathrm{cut}}, \end{cases}$$



where ϵ, σ and r_{cut} are user-defined parameters. Since the overlap energy is at most ϵ, we do need force capping for this potential.

Suggested exercises

1. calculate the correspond force expression. Write the force in the usual form
$$F(\|r\|)/\|r\| \times \vec{r}.$$
2. implement the Gaussian potential following the receipe below.
3. To check your implementation, it is a good idea to make sure that the force and energy of your potential are consistent. Do this using `compare_potential_and_forces.tcl` from the `samples` folder.

4. now use `liquid.tcl` provided with this tutorial in order to calculate the radial distribution function at various densities using your new potential and the Lennard-Jones potential. What do you expect? Where should be differences to the Lennard-Jones potential? Does your potential meet your expectations?

Adding a new potential:

- use the hertzian potential as a template
- potential and force calculation: `add_gaussian_pair_force` and `hertzian_pair_energy` in `gaussian.h`
- set parameters: `hertzian_set_params` in `gaussian.c` and `gaussian.h`
- make the parameters exist: `struct IA_parameters` in `interaction_data.h`
- integrate with interactions: `interaction_data.c`
  - include header `gaussian.h`
  - initialize and copy parameters: `initialize_ia_params`
  - make cutoff known: `recalc_maximal_cutoff_nonbonded`
- integrate with force and pressure: `calc_non_bonded_pair_force_parts` in `forces.h`
- integrate with energy calculation: `calc_non_bonded_pair_energy` in `energy.h`
- parse and write the parameters: `tclcommand_inter_parse_gaussian` and `tclprint_to_result_GaussianIA` in `tcl/gaussian_tcl.c` and `tcl/gaussian_tcl.h`
- add to the interaction parser: `tclcommand_inter_parse_non_bonded` (macro REGISTER_NONBONDED) and `tclprint_to_result_NonbondedIA` in `tcl/interaction_data_tcl.c`
- add `gaussian.c`, `gaussian.h`, `tcl/gaussian_tcl.h` and `tcl/gaussian_tcl.h` to the build system: `src/Makefile.am`
- add `GAUSSIAN` to the config system: `features.def`
- document Gaussian potential: `doc/ug/inter.tex`
- use `bootstrap.sh` to update the build system for the new potential, and recompile.

3 Committing your new potential

Now it is time to make your new potential available to the ESPResSo community.

- Check the output of `git status` to see what you changed. The modified files already appear as such (but not staged for commit yet!), the new files appear as untracked.

- use `git add <file>` to add the new and modified files to the next commit, and commit your changes using `git commit`.

- use `git format-patch HEAD^` to create a patch file suitable to be mailed. Check that the patch looks again, and send it to us to include your code in the next release!