

## GROMACS - Feature #2791

### Implement a module for lambda-dynamics simulations (lambda\_site module)

12/04/2018 11:54 PM - Thomas Ullmann

<b>Status:</b>	In Progress
<b>Priority:</b>	Normal
<b>Assignee:</b>	Thomas Ullmann
<b>Category:</b>	mdrun
<b>Target version:</b>	future
<b>Difficulty:</b>	uncategorized

#### Description

#### Purpose

This issue gathers information about the ongoing development of a lambda-dynamics module for GROMACS to enable constant-pH and more generally constant-chemical potential simulations and free energy calculations with multiple end states.

#### Introduction

Lambda-dynamics is a way of allowing for interconversion of a simulation system between distinct chemical states, for example between different protonation states. The system is partitioned into the invariable (fixed topology and partial charges) so-called background and a number of variable sites, e.g., protonatable sidechains of a biopolymer or redox-active cofactors. A variable site can occur in multiple distinct forms which can differ in charge distribution and/or topology. These forms could for example correspond to the different protonation forms of a titratable sidechain. lambda-dynamics allows for a continuous interconversion between the forms of a site during the simulation in response to the energy difference between the site forms. These energy differences comprise energy contributions intrinsic to the site and interactions of the site with its environment, including interactions with other variable sites.

A site can have  $n$  forms. Each site form has an associated lambda coordinate. If the lambda coordinate approaches a value of 1.0, the site form is fully present, if the coordinate approaches a value of zero, the site form is fully absent. The sum over all  $n$  lambda variables of a site should ideally be equal to 1 (you want to have exactly one copy of the site in the system, no more and no less). In lambda-dynamics, the lambda variables are perceived as coordinates of a fictitious particle, the so-called lambda particle with a fictitious mass. The lambda particle can move along the lambda coordinates like the normal particles along their Cartesian coordinates.

The permitted range of lambda values  $0 \leq \lambda \leq 1$ , and the constraint on the sum of the lambda values of a site is ensured by introducing an auxiliary set of coordinates, the so-called simulation coordinates, in which the actual evolution of the system during the simulation takes place. Different lambda dynamics variants with different sets of simulation coordinates exist. After each position update, the simulation coordinates are mapped back to the lambda coordinates for the next time step.

The forces that govern the motion of the lambda-particles along the simulation coordinates (and hence along the lambda coordinates) are given by the partial derivatives of the energy function with respect to the simulation coordinates. These partial derivatives can be computed via the chain rule of differentiation from the partial derivatives of the energy function with respect to the lambda coordinates and the partial derivatives of the lambda coordinates with respect to the simulation coordinates. The precise mathematical functions involved differ between different lambda-dynamics variants. In the lambda\_site module, these variants are represented by implementation variants of the polymorphic `LambdaSiteInternals` object.

The computationally most intense part of the simulation is the calculation of electrostatic interactions that contribute to the forces acting on the normal simulation particles and those acting on the lambda-particles. There are different methods for their computation most prominently the Particle-Mesh-Ewald (PME) method and the Fast Multipole Method (FMM). A FMM which offers support for lambda-dynamics is currently developed by Ivo Kabadschow, Andreas Beckmann, Holger Dachsel and coworkers at the Juelich Supercomputing Centre. `fmsolvr` will be hosted [here](#).

Typically one adds a bias potential to encourage the system to adopt lambda values close to 0 or 1 (physical states) and spend less time in unphysical intermediate states. Different variants of such a bias potential are represented by the polymorphic `LambdaSiteBiasPotential` object.

The sites of a form may have a differing net charge. Interconversion between different site forms may thus lead to a change in the net charge of the simulated molecule(s). It has been recognized in the literature (see [this paper](#) by Gerrit Groenhof and coworkers and contemporary papers of Roux, Huenenberger and Rocklin) that a non-zero net charge of the system in simulations under periodic boundary conditions can lead to simulation artifacts. Thus, the change in the net charge of the simulated molecules should be counterbalanced by adding or removing mobile ions in the surrounding solution. Within lambda-dynamics, this balancing can be achieved by treating the mobile ions as variable sites that can interconvert between a neutral form (e.g., water) and a site form with

the normal ion charge. The lambda coordinates of the biomolecule and the counterions are coupled by a constraint that ensures a neutral overall net charge of the system. The constraint is implemented in the LambdaSiteConstraints object following the [LINCS](#) formalism. This [solution](#) is a natural extension of the widely adopted practice of neutralizing the net charge of a conventional simulation system with an appropriate number of counter-ions.

We want to simulate a model of a (bio)molecular system with a number of variable sites, most often protonatable sites. Each variable site can interconvert between different forms that may possess different net charge. Overall charge neutrality of the system is ensured by coupling the interconversion of the site forms to the appearance and disappearance of counter-ions with a constraint. The (bio)molecule is depicted in gray with its variable sites in gray. The positively and negatively charged mobile counterions are depicted in blue and red, respectively.

## Overview of Related Patches

The following figure shows the currently submitted or planned patches. An arrow pointing from patch A to patch B indicates that patch B requires patch A. The nodes of already submitted patches are clickable and link to the corresponding patch set in Gerrit (open image in a new tab). The figure is meant to be updated during the development to keep track of the related patch sets.

stage1.dot.svg

## Related Issues

Issue [#1332](#) "Supporting multiple end states instead of just A and B"

Issue [#1652](#) "Decide how to represent multiple lambda states internally"

Issue [#1653](#) "Decide how to represent multiple lambda states in the .top file and how to parse them"

Issue [#1654](#) "How to carry out movement between chemical end states in a multiple end state framework?"

Issue [#1658](#) "Electrostatics treatment for multiple lambda sites"

---

## History

**#1 - 12/05/2018 03:21 PM - Carsten Kutzner**

- Status changed from New to In Progress

**#2 - 02/08/2019 07:26 PM - Gerrit Code Review Bot**

Gerrit received a related patchset '67' for Issue [#2791](#).

Uploader: Thomas Ullmann ([thomas.ullmann@mpibpc.mpg.de](mailto:thomas.ullmann@mpibpc.mpg.de))

Change-Id: gromacs~master~1d5b32ac18479b32ddb6e7d8f16b829c468b66e6

Gerrit URL: <https://gerrit.gromacs.org/6801>

**#3 - 03/01/2019 01:36 AM - Gerrit Code Review Bot**

Gerrit received a related patchset '1' for Issue [#2791](#).

Uploader: Thomas Ullmann ([thomas.ullmann@mpibpc.mpg.de](mailto:thomas.ullmann@mpibpc.mpg.de))

Change-Id: gromacs~master~1e36cb0a2f31f2f2768e0c55fa864f97fe5b9af24

Gerrit URL: <https://gerrit.gromacs.org/9253>