

GROMACS - Feature #1332

Supporting multiple end states instead of just A and B

09/06/2013 04:20 PM - Michael Shirts

Status:	In Progress	
Priority:	Normal	
Assignee:	Michael Shirts	
Category:	mdrun	
Target version:		
Difficulty:	uncategorized	
Description		
<p>There are a number of reasons to be able to have more than one end state. There are lots of potential applications, but will require some careful thinking and code reorganization. Gerrit (Groenhof, not the code verification system) and I recently discussed it briefly again.</p> <p>This is not a priority now (soonest I would imagine is 5.1) , but I wanted to create this issue so that we have a place to put this discussion. I didn't see it anywhere else.</p> <p>I think that the best way to do this in topologies is instead of stacking new parameters vertically repeat parameter sets</p> <pre>[atoms] 1 types A types B parameters A Parameters B 2 types A types B parameters A Parameters B [atoms] 1 types A Parameters A 2 types A Parameters A 1 types B parameters B 2 types B parameters B</pre> <p>This will scale to as many end states are needed.</p> <p>The internals are more complicated, and should be discussed later. Most of the changes are obvious; some are not as obvious.</p> <p>I look forward to much interesting discussion, especially after Dec 1 :)</p>		
Subtasks:		
Feature # 1652: Decide how to represent multiple lambda states internally		New
Feature # 1653: Decide how to represent multiple lambda states in the .top file and how...		New
Feature # 1654: How to carry out movement between chemical end states in a multiple end...		New
Feature # 1658: Electrostatics treatment for multiple lambda sites		New

History

#1 - 09/06/2013 04:39 PM - Michael Shirts

- Priority changed from Normal to Low

#2 - 09/06/2013 08:14 PM - Thomas Ullmann

I started working on such a solution for lambda-dynamics and free energy applications, too. The proposed solution allows for single-topology, dual topology and hybrid setups. I already talked about this variant with Gerrit. He noted a potential pitfall (uncoupled atoms of the dual topology part drifting through the system), but that issue is separate from how to define the topology. I also think that the problem could possibly be solved even without restraint potentials or fixed-distance constraints.

Below is a more detailed description of the intended topology

Please let me know what you think.

A residue/site can have any number of forms (protonation forms etc.), which could also be used to define the end states of a free energy calculation.

Each atom can either belong to the single-topology part (same atom coordinates in each form) or to one particular form in the dual topology part (separate atom coordinates).

Bonds can only be formed between:

- two atoms belonging to the single-topology part
 - two atoms belonging to the same form of the dual topology part
 - an atom of the single-topology part and an atom of a particular form of the dual topology part
- but not between two atoms from different forms of the dual topology part

Each atom of the single-topology part has to be assigned either as many atom type-partial charge combinations as there are forms or a single such combination, which would mean that these parameters are assigned to the atom in all forms. Currently, I envisioned a horizontal arrangement of the parameter sets, but that could be changed. A horizontal arrangement might however be easier to read and one can retain a consistency check for unique atom names within the single-topology part. Programming-wise, the horizontal arrangement should be fairly easy to implement. I already have something similar for a different purpose using C++ stream operators. Such a solution is very concise in the source code and easy to understand.

A form of the dual topology part is marked in the .rtp file by a "sub-block" opened by [form] and closed by [form-end]. Multiple forms can be listed consecutively. The order of the listing has to comply with the order of the forms in the single-topology part. The number of forms must be the the same in single-topology and dual topology parts. That is, one can either define as many form blocks as there are forms in the single-topology part or no dual-topology parts at all. Form blocks can also be empty.

A residue topology is also valid if there are no atoms belonging to the single-topology part or if all or some of the atoms in the single-topology part are listed with just a single atom type-charge pair (would implicitly mean that the parameters are the same in all forms).

Atom names can only be used once within the single topology part. Atom names used in the single-topology part are not allowed again in the dual-topology part. Atom names can only appear once per form in the dual topology part. The same atom name can appear in the dual topology part of multiple forms.

My current idea of how a residue topology with multiple forms/end states could look like:

```
-----
[ ASP ]
[ atoms ]
; single topology part: same atom coordinates but possibly different
;     interaction parameters (partial charges and
;     atom types/Lennard-Jones parameter sets)
;     For each atom, either specify a single parameter
;     set or one parameter set for each residue form.
;     ASPD  ASP1S  ASP1A  ASP2S  ASP2A
; atom_name atom_type_1 partial_charge_1 {atom_type_2 partial_charge_2 ...} charge_group
OD1  OC -0.76 OH1 -0.61 OH1 -0.61 OB -0.55 OB -0.55 8
; ... more atoms ...

; dual topology part:
[form]
```

```
; atom_name atom_type partial_charge charge_group
HD11      H      0.44      12
; ... more atoms ...
```

[end-form]

```
; four more form blocks following
-----
```

#3 - 09/07/2013 04:22 AM - Michael Shirts

Thanks for the comments (though my responses might not be that detailed for month or two)! This sounds interesting. We do want a solution that support multiple topologies with molecules as well as .rtp's

#4 - 06/09/2014 08:51 PM - Erik Lindahl

- Target version changed from 5.x to future

#5 - 08/02/2014 10:59 PM - Michael Shirts

- Target version changed from future to 5.x

Coming back to this; I talked a little with Mark about what would need to be done to make this work.

I think it's pretty straightforward in terms of STORING the extra topologies -- everywhere that there are A and B states in the current topology, we replace with an array instead, with dimensions determined at topology creation time (if only A state, then only the [0] state is filled.

I think what is less clear is how do handle some of the bookkeeping. The main questions are

- 1) how to encode the extra topologies in a file (lots of possibilities, not clear which is easiest/best)
- 2) how to manage the energy information that is produced from having multiple end states (more complicated)

As examples of problems with 2):

- Are there multiple lambda vectors, controlling where one is with respect to each lambda state?
- If so, how do these lambdas get defined in the .mdp?
- Are these from the same root state (A->B, A->C, A->D), consecutive (A->B, B->C, C->D), or pairwise (all N choose 2 pathways between N endstates?)
- Is it possible to be halfway from A to B and halfway from B to C? i.e. do N intermediates define an N-dimensional space, and one can be anywhere in that space, or can the system only be along edges?

I think that to resolve the degeneracies in possibilities, we need to be a bit more specific about what one would want to do with multiple end states. That discussion may guide how multiple end states would be implemented.

Reasons I can think of

- Constant PH or constant-chemical-potential-of-tautomers simulations; often requires multiple protonation states.
- Computing the free energy of multiple ligand simultaneously
- Scanning multiple side chains simultaneously.
- EDS (enveloping distribution sampling) -- Niels Hanson (formerly van Gunsteren Group, now assistant professor) is thinking of implementing this.

Can we brainstorm other applications that people would use multiple end states for for to help understand what constraints we would have on the implementation?

#6 - 08/04/2014 07:44 AM - Gerrit Groenhof

- Status changed from New to Closed

> Can we brainstorm other applications that people would use multiple end states for for to help understand what constraints we would have on the implementation?

- Empirical Valence Bond

#7 - 08/04/2014 07:45 AM - Gerrit Groenhof

- Status changed from Closed to In Progress

#8 - 10/13/2015 08:52 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1332](#).

Uploader: Thomas Ullmann (thomas_ullmann@web.de)

Change-Id: I68814357b5959ea691bb508fd13aeb8eba4f1fff

Gerrit URL: <https://gerrit.gromacs.org/5204>

#9 - 10/27/2015 06:18 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1332](#).
Uploader: Thomas Ullmann (thomas_ullmann@web.de)
Change-Id: lc80a3e9de7c54440a2734a6d798479c452383e5e
Gerrit URL: <https://gerrit.gromacs.org/5250>

#10 - 07/11/2016 08:15 PM - Mark Abraham

- Target version deleted (5.x)

#11 - 07/24/2017 12:11 PM - Gerrit Code Review Bot

Gerrit received a related patchset '5' for Issue [#1332](#).
Uploader: Thomas Ullmann (thomas_ullmann@web.de)
Change-Id: gromacs~master~lc80a3e9de7c54440a2734a6d798479c452383e5e
Gerrit URL: <https://gerrit.gromacs.org/5250>

#12 - 06/05/2018 03:58 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1332](#).
Uploader: Thomas Ullmann (thomas_ullmann@web.de)
Change-Id: gromacs~master~lb042563a6df9b275a6e349a2b5c7434585ac32ac
Gerrit URL: <https://gerrit.gromacs.org/7978>