### GROMACS - Feature #1867

**make coupling implementations reversible**

12/02/2015 07:43 AM - Mark Abraham

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>uncategorized</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>uncategorized</td>
</tr>
</tbody>
</table>

**Description**

Our current implementations of all thermostats and barostats (code in update.cpp and coupling.cpp) compute a scaling factor (or extended-variable velocity) at nsttcouple/nstpcouple steps, and use that at each subsequent step. Thus, the implementation of these coupling modes is irreversible, along with various other known defects. (Perhaps our GROMOS heritage is still showing here, as with the former non-Trotter twin-range scheme.) The verlocity-Verlet implementations require periods of 1 step, so do not have this problem.

I think all these implementations should be impulsive, and thus reversible - do the rescaling at nsttcouple/nstpcouple steps, scaled by the period in Trotter style, then evolve at constant E. I think that if someone really wants smoother velocity variation as well as coupling, then they should choose a coupling period short enough to suit them (and pay for the extra global communication), rather than have all the default leap-frog integration schemes be slightly non-physical.

That would be a behaviour change, which I would suggest we implement by

- introducing `.mdp` option nstcoupling (default 10; VV integrators again requiring 1)
- removing nsttcouple, nstpcouple, nstcalcenergy and mdrun -gcom
- requiring nstenergy, nstlog, nstdhdl, nstexpanded, nstcomm, and replica-exchange periods to be multiples of nstcoupling
- making grompp reject old `.mdp` coupling options, so users can be prompted to choose the coupling behaviour they want
- making mdrun reject old `.tpr` files unless they have nsttcouple == nstpcouple, and that value a factor of nstcalcenergy and nstenergy (implemented with nstcoupling set to that value) and satisfy the other constraints - thus a large majority of old `.tpr` files will still run, but can issue a note about the changed implementation of coupling

This is also considerably simpler to implement in `do_md()`, because we can no longer have different frequencies for different coupling (which was probably untested), can't override the model physics with a command-line option (also not currently automatically tested), and contributes to having many fewer combinations of things for which to consider adding tests (also helped by nstcalclr and nstmultisimsync going away). Naturally, that would need automating of a lot of long-overdue mdrun quality testing, but making that testing problem less intractable is attractive.

Note that nstlist and any autotuning will have nothing to do with this (but historically nstlist probably did because of the twin-range scheme).

Thoughts?

**Related issues:**

- Related to GROMACS - Feature #1137: Proposal for integrator framework (do_md)... - New

**History**

#1 - 02/26/2018 04:03 PM - Mark Abraham

- Related to Feature #1137: Proposal for integrator framework (do_md) in future GROMACS added

02/19/2020