GROMACS - Task #1868

implement mdrun -rerun better, simplifying do_md

12/02/2015 08:34 AM - Mark Abraham

Status:	Closed	
Priority:	Normal	
Assignee:	Mark Abraham	
Category:	mdrun	
Target version:	2019	
Difficulty:	uncategorized	

Description

I think the main uses for mdrun -rerun are to compute PE-like quantities from the particle positions against the Hamiltonian in the .tpr. This Hamiltonian might be different from that used in the simulation, or possibly treats only a subset of the system, or with PE decomposed into groups. Currently mdrun -rerun accepts the .tpr and does its best not to segfault, but it writes a large amount of numbers that are either junk, or not a "rerun" even if you use the same .tpr.

Full reproduction of the results of a step with mdrun -rerun requires the velocity update to be reproducible, because trajectories are written with x leading v (even for on-step integrators like velocity Verlet), so one needs a velocity update to compute velocity-based quantities such as KE, T, P, conserved E, etc. This is tricky because

- rerun can't reproduce the velocity update for extended ensemble methods (Nose-Hoover, Parrinello-Rahman, MTTK) because these use extended-ensemble dynamical quantities such position and momentum of the extended variables that are not available from a trajectory (though they are often in .tpr and always in .cpt)
- rerun can't reproduce any of the non-extended leapfrog thermostat or barostat velocity updates, unless nsttcouple==1 and/or nstpcouple==1, because the implementation computes a scaling factor at coupling steps, and then applies it at every intervening step, and that's not saved to the trajectory
- the use of nsttcouple and nstpcouple mean that only the step number lets mdrun tell if the rerun velocity update should be associated with coupling even if the data was present - not a big deal in itself, because the user can't expect a rerun if they changed the step numbers
- stochastic integrators (and also Bussi and Andersen thermostats) can re-run only if the random numbers are drawn correctly - this is probably
  done right already, if the step number is present (and unchanged)
- rerun can't reproduce the v-rescale conserved quantity, which requires the integral of changes in KE over all steps, and the frames being rerun cannot be assumed to have any such property
- the use of mdrun -gcom makes reproducing a velocity update nearly impossible because we don't know afterwards whether it was used unless we inspect the .log file

Already we hack nstglobalcomm = 1 for rerun, and we could make something work also by requiring all the coupling periods to be 1 (or requiring such a .tpr), but it's not now a rerun, and enforcing this doesn't have a clear purpose that I know of.

So, a rerun should simply evaluate potential energy and forces, report those, and not pretend it can sensibly do anything else. If so, then we can extract a much simpler implementation of do_rerun() out of do_md(), and thereby simplify do_md().

If the rerun .tpr is NVE molecular dynamics (any integrator), or probably the various stochastical dynamics, then the velocity update is trivial and such a rerun could report things derived from velocities, e.g. computed from a new Hamiltonian, but does this serve any purpose? Or can one be imagined?

Related issues:

Related to GROMACS - Task #1793: cleanup of integration loop	**New**	
Related to GROMACS - Bug #2244: Test rerun with FEP	**Closed**	
Related to GROMACS - Bug #2112: Rerun tests with MSVC may expose existing bug	**Closed**	
Related to GROMACS - Task #1246: expanded ensemble .tpr cannot be rerun	**New**	05/10/2013
Related to GROMACS - Task #2569: announce deprecations in GROMACS 2019	**Closed**	
Related to GROMACS - Bug #2849: Free energy discrepancies between GROMACS ver...	**Closed**	

Associated revisions

Revision 280ba5a3 - 03/28/2018 01:49 PM - Mark Abraham

Free more memory in grompp and mdrun

This will be increasingly useful as we start adding more GoogleTest

cases that call grompp and mdrun repeatedly.

Refs #1793, #1868

Change-Id: Ic0faaa4c3dec2891d0cdc6166561f8ccd4391f44

Revision 8ee76066 - 04/04/2018 12:51 PM - Pascal Merz

Move checkNumberOfBondedInteractions

In task #1868, we will move the rerun functionality from do_md to
a separate function in a new file. As both do_md and the new
do_rerun need access to checkNumberOfBondedInteractions(), it needs
to be moved to a more central location.

Change-Id: Iae0ace11db2f83e5034fdfae21ccac3e7b9e34d1

Revision c7411433 - 04/16/2018 09:17 AM - Mark Abraham

Updated mdrun comparison infrastructure

The functionality for running grompp to produce a .tpr file useful for
comparing two mdrun calculations works better as free functions than a
base class. This patch is essentially a rewrite of the interface,
though the core functionality is roughly the same.

Used raw string literals for nicer looking embedded strings.

Refs #1868

Change-Id: I4b5137b34c3da286e344809b71afd985d9c7bfab

Revision b70481d1 - 08/21/2018 09:59 AM - Mark Abraham

Re-implemented mdrun -rerun tests

Used the new infrastructure to let the rerun test check that the
energies and trajectories (including forces) produced by a rerun
actually match those produced by a normal mdrun. Such a test running
in parallel would have prevented #1868, and will help checking that
any changed or new integrator functionality does the job that it
should do.

Updated some infrastructure to not care whether it gets passed
a C or C++ string.

Removed small pieces of infrastructure originally intended for use by
this patch but which are no longer required.

Refs #1868, #2112

Change-Id: Ifcb211aba62ccb79ebaec21a3088c9e60a618c86

Revision 564a67c0 - 09/21/2018 10:08 AM - Pascal Merz

Make CGLO_CHECK_NUMBER_OF_BONDED_INTERACTIONS require global communication

Currently, giving the CGLO_CHECK_NUMBER_OF_BONDED_INTERACTIONS flag, but
none of CGLO_TEMPERATURE, CGLO_STOPCM, CGLO_PRESSURE, CGLO_CONSTRAINT, or
CGLO_ENERGY does not prompt global_stat() in compute_globals(). This has
currently no implications because this specific flag is never set without
at least one of the others, but it is not intuitive and becomes a problem
when setting up a rerun without temperature calculation. This change fixes
this.

Refs #1868, #2647

Change-Id: I53b4dd9a1b7b273f1ce99e8e8c78bf386a693842

Revision 8270c604 - 09/25/2018 02:37 AM - Pascal Merz

Prepare rerun commit

In Task #1868, we will break out rerun from the do_md code. This
results in new files, rerun.{h,cpp}, containing a do_rerun function

which includes only the portions of code in do_md which are relevant
for rerun. It does, however, make sense to diff this new function to
the current do_md to be able to review the changes made. This commit
therefore duplicates the current md.{h,cpp} files to allow for easier
reviewing of the following changes. It does also change runner.cpp to
send any rerun through the new code path. This allows to create to
independent follow-up commits, namely
1 Cleaning up do_md() [I17c9ae4c1c88bdb328e17e75fec55a558613cff0], and
2 Cleaning up do_rerun() [I522d56a610f2a6ff189484c9311b6f53c775ed47].
Commit 1 will thereby likely be easy to review and merge, while
commit 2 might require more discussion.

Refs #1868

Change-Id: I832ce07e3c5932a42151983deb8f527f988b2a2e

Revision c52e6838 - 09/25/2018 07:32 AM - Pascal Merz

Remove unnecessary functionality from do_rerun

As discussed in task #1868, the rerun functionality gets reduced to essentially
"take a configuration and calculate forces and energies". If the trajectory contains
velocities, they are ignored. This allows to remove a lot of functionality from
do_rerun, which is done in this commit.

Functionality removed includes

- Reading of velocities in reruns (reruns won't contain kinetic energy)   * Constraints   * Temperature / pressure coupling   * Essential dynamics   * Interactive MD   * Simulated annealing   * Multisimulations   * AWH   * Expanded ensemble   * Replica exchange   * Checkpointing   * Counter resetting (no benchmarking)   * PME tuning   * FAHCORE   * EnergyHistory

The following quantities are not reported:

- Kinetic energy   * Total energy   * Conserved energy   * Temperature(s)   * Pressure, pressure tensor, and pressure dispersion correction   * Virial tensor(s)

This change also changes the rerun tests not to test for correct
reproduction of the velocities (since they are intentionally set
to zero), and adds some information to the documentation.

Refs #1868

Change-Id: I522d56a610f2a6ff189484c9311b6f53c775ed47

Revision 11c0af7b - 09/25/2018 08:33 AM - Pascal Merz

Remove rerun from do_md

As the rerun functionality is handled in a separate function now, this
commit removes rerun from do_md.

refs #1868

Change-Id: I17c9ae4c1c88bdb328e17e75fec55a558613cff0

History

#1 - 12/11/2015 08:50 AM - Mark Abraham

Also, bd + constraints gets KE (and thus related quantities) slightly wrong for the first frame of a rerun (only).

#2 - 06/01/2016 02:41 PM - Mark Abraham

- Category set to mdrun

- Target version set to 2018

#3 - 10/03/2016 09:30 PM - Gerrit Code Review Bot

Gerrit received a related patchset '13' for Issue #1868.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: I7f99eba36184f564108c38865fe219f1bca55354
Gerrit URL: https://gerrit.gromacs.org/5435

#4 - 01/31/2017 05:13 AM - Michael Shirts

The only thing that I think of where velocities in rerun were important is to calculate things like transport properties, with different Hamiltonians. Such a think can be done (http://aip.scitation.org/doi/abs/10.1063/1.3592152), and I'd like to do it eventually.

However, the only thing that changes the contribution to the Boltzmann weight is the mass, and that can be handled analytically (the KE scales linearly with them mass). So it can be done with the original trajectory.

So, I think the conclusion is -- explicitly ditch the velocities (perhaps fill with zeros, and print lots of warnings).

If we change our mind, I think we will have to have special rules for generating the velocities, which should probably be done by different code anyway, and that can be added in later.

#5 - 01/31/2017 03:01 PM - Mark Abraham

OK, sounds good. Any thoughts Berk (or others)?

#6 - 01/31/2017 03:02 PM - Mark Abraham

- Related to Task #1793: cleanup of integration loop added

#7 - 01/31/2017 03:05 PM - Gerrit Code Review Bot

Gerrit received a related patchset '20' for Issue #1868.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~I7f99eba36184f564108c38865fe219f1bca55354
Gerrit URL: https://gerrit.gromacs.org/5435

#8 - 03/15/2017 08:05 PM - Mark Abraham

- File rerun.bundle added

Following on recent discussion with Michael & colleages, attached is some very crude WIP branches from my attempts to separate rerun from do_md. Do git clone rerun.bundle rerun.

A bunch of things have been cleaned up since then, so I strongly suggest using the contents as a guide, rather than trying a rebase or anything.

The major idea is to do a copy-paste, and then delete all the code in branches that become dead because rerun is now known to be on or off. Compilers are somewhat helpful at that these days, so you often go through a series of "oh ok remove everything related to this".

#9 - 06/27/2017 11:13 PM - Mark Abraham

- Target version changed from 2018 to 2019

We have WIP, but not for 2017

#10 - 02/05/2018 05:50 PM - Mark Abraham

- Related to Bug #2244: Test rerun with FEP added

#11 - 02/05/2018 05:50 PM - Mark Abraham

- Related to Bug #2112: Rerun tests with MSVC may expose existing bug added

#12 - 03/06/2018 11:30 PM - Mark Abraham

- Related to Task #1246: expanded ensemble .tpr cannot be rerun added

#13 - 03/13/2018 07:41 AM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~I523cbdc601354beeb62277e598bf23a5b5de3951
Gerrit URL: https://gerrit.gromacs.org/7668

#14 - 03/13/2018 01:24 PM - Gerrit Code Review Bot

Gerrit received a related patchset '2' for Issue #1868.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~Ic0faaa4c3dec2891d0cdc6166561f8ccd4391f44
Gerrit URL: https://gerrit.gromacs.org/7671

#15 - 03/19/2018 04:51 PM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)

Change-Id: gromacs~master~I062a7551b794419e15174e93d75e10938802b2ff
Gerrit URL: https://gerrit.gromacs.org/7692

#16 - 03/19/2018 04:54 PM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~I575d9e0b55c727572c2f016190569a68c17d2d0c
Gerrit URL: https://gerrit.gromacs.org/7693

#17 - 03/19/2018 07:37 PM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~Ieb9cdab8d82565e9efbfd70d6c013f810f12a199
Gerrit URL: https://gerrit.gromacs.org/7694

#18 - 03/19/2018 08:10 PM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~Iae0ace11db2f83e5034fdfae21ccac3e7b9e34d1
Gerrit URL: https://gerrit.gromacs.org/7695

#19 - 03/27/2018 06:21 AM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~I17c9ae4c1c88bdb328e17e75fec55a558613cff0
Gerrit URL: https://gerrit.gromacs.org/7723

#20 - 03/27/2018 06:55 AM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~I522d56a610f2a6ff189484c9311b6f53c775ed47
Gerrit URL: https://gerrit.gromacs.org/7724

#21 - 03/28/2018 03:29 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1868.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~I4b5137b34c3da286e344809b71afd985d9c7bfab
Gerrit URL: https://gerrit.gromacs.org/7735

#22 - 03/28/2018 03:35 PM - Mark Abraham

Note to self - remember Michael's comments at the end of https://gerrit.gromacs.org/#/c/5435/

#23 - 03/29/2018 04:31 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1868.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~Ifcb211aba62ccb79ebaec21a3088c9e60a618c86
Gerrit URL: https://gerrit.gromacs.org/7742

#24 - 04/06/2018 06:01 PM - Mark Abraham

- Status changed from New to In Progress

#25 - 04/18/2018 11:38 PM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~I832ce07e3c5932a42151983deb8f527f988b2a2e
Gerrit URL: https://gerrit.gromacs.org/7778

#26 - 09/13/2018 01:47 AM - Mark Abraham

- Related to Task #2569: announce deprecations in GROMACS 2019 added

#27 - 09/21/2018 12:37 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1868.
Uploader: Pascal Merz (pascal.merz@colorado.edu)

Change-Id: gromacs~master~I53b4dd9a1b7b273f1ce99e8e8c78bf386a693842
Gerrit URL: https://gerrit.gromacs.org/8405

#28 - 09/21/2018 02:04 PM - Eric Irrgang

The commit message in https://gerrit.gromacs.org/c/7724/ describes an automatic decision of whether or not to report the virial based on whether there are constraints in the topology. Is this better than consistent output? (e.g. not reporting the virial, or making the virial a (sanity-checked) option)

This sounds like complex behavior that could be hard to deal with down the line... Is there a clear case in which a user or API client would want a different style of output based on their topology instead of the parameters they have provided? Maybe just don't report the virial unless explicitly requested (with an error if not possible). Preserving old behavior for some inputs and not others sounds problematic. But if it is valuable, maybe report the virial as the default behavior of an input parameter, with an error that the input data is not compatible with the task as specified if constraints are present and virial is requested. That behavior could be deprecated with a change in default behavior (encourage explicit parameters) in a future version.

#29 - 09/23/2018 06:34 AM - Michael Shirts

> Maybe just don't report the virial unless explicitly requested (with an error if not possible). Preserving old behavior for some inputs and not others sounds problematic.

I think this is probably better, to have the 'vanilla' information printed (i.e. not virial) with a way to print stuff that could possibly useful (i.e. virial in the case of no constraints). It's already an advanced request, since you would need to calculate the effective pressure by analytically including the kinetic energy, and knowing you set something up without constraints.

I would also suggest warnings why things are not being printed out, so it makes sense for users, and perhaps listing the options in the warning that one COULD use to get the information, in the cases it might still be available).

#30 - 09/24/2018 10:42 AM - Mark Abraham

Agree on the potential for confusion from sometimes calculating things, and that the user guide should add advice about what is happening and why, and if there are alternatives. Log file can direct the user to that.

If someone comes up with a good use for a virial, then with the change to using gmx rerun then we have liberty to consider command line options there that would have been too complicated when we had to use gmx mdrun -rerun.

#31 - 10/03/2018 09:40 PM - Mark Abraham

- Status changed from In Progress to Resolved

rerun has been moved out of do_md, so the main task here is complete

#32 - 10/03/2018 09:40 PM - Mark Abraham

- Status changed from Resolved to Closed

#33 - 02/07/2019 05:53 PM - Mark Abraham

- Related to Bug #2849: Free energy discrepancies between GROMACS versions added

Files

rerun.bundle	43.1 KB	03/15/2017	Mark Abraham