GROMACS - Task #2943

Rename Integrator

05/08/2019 08:39 AM - Pascal Merz

Status:	Closed	
Priority:	Normal	
Assignee:	Pascal Merz	
Category:	mdrun	
Target version:	2020	
Difficulty:	simple	

Description

In the current code, the name *Integrator* is used for the class of the object owning the family of functions running some type of simulations, i.e. do_md, do_rerun, do_steep, do_cg, do_lbfgs, do_nm, do_tpi, and do_mimic. It is named after the mdp-option integrator. There are two problems with this naming:

1. Most of these functions don't actually do any integration, and as new methods are implemented (e.g. Monte Carlo or MC/MD-hybrid methods), this is not improving.
2. There is potential for confusion between the current concept of Integrator as the complete description of a simulation run and the actual propagation of the micro-state (e.g. leap-frog *integrator*, velocity-verlet *integrator*, ...)

Based on discussions with several developers, I would therefore suggest to rename the current Integrator class and the corresponding mdp-option, as well as to stop using *integrator* for any current or future implementations of the micro-state propagator.

For the propagation of the micro-state, the term **propagator** seems appropriate, and allows to also accommodate non-dynamical ways of propagating the system (e.g. minimization, MC steps, ...). Besides the propagation of positions and velocities (including constraining), this should probably also include functions to calculate forces, energies, virial, as their placement in the instruction flow is dependent on the propagation method chosen.

The prescription of the simulation (formerly known as Integrator) is a bit harder to rename. It contains the components of the propagator, but also plenty of plumbing for propagator stages, parallelization, I/O, ... Candidates include Simulation (as any run of GROMACS can probably be classified as (molecular) simulation), Program, Prescription, Driver, or a combination of these (e.g. SimulationPrescription).
Some thoughts on this (credit to Mark for this):

- Schedule is currently being used for the set of work within the force calculation plus its reduction, but Prashanth (or Mark?) also suggested recipe - something about schedule was not good, probably on gerrit somewhere
- Module is already used for MDModule like ElectricField, gmx command-line module like gmx grompp, and source-code unit like src/gromacs/swap, so avoid that
- The task graph that implements the do_md level thing has the components of the Propagator as part of it, but also plenty of plumbing for propagator stages and I/O
- Runner is currently used for the thing that coordinates setting up for and dispatching the do_md level thing, probably best to leave that alone
- Context, Session, Workflow are all used by gmxapi, probably best avoided
- Context, program, driver, and command also have a GPU-related meaning, probably best avoided
- Pipeline is too linear

This leaves something like **Simulation**, **Simulator**, **SimulationPrescription** as most likely candidates. I'd be happy to hear other suggestions and opinions!

Associated revisions

Revision 66bf1a79 - 05/25/2019 08:25 PM - Pascal Merz

Rename Integrator class

This renames the `Integrator` class to `Simulator`, which does not
actually represent integrators, but rather types of simulations
(some of which contain an integrator, but not all!).

Refs #2943

Leaves the TODO to rename the corresponding mdp-option, but no
consensus on the new name has been reached yet.

Change-Id: I9bc3158856cd68dfc2ca5f3da793cd1ecbc2ffaf

History

#1 - 05/08/2019 08:40 AM - Pascal Merz

(My favorite would probably be **SimulationRunner**, but this seems a bit close to the runner we already have one level above...)

#2 - 05/08/2019 02:51 PM - Berk Hess

But something like normal mode analysis is not a simulation, minimization probably also not. So I would avoid the word simulation. I still think something with "propagator" is better. But maybe there is something even better.

#3 - 05/08/2019 04:08 PM - Eric Irrgang

gmxapi use of the term "Session" is more consistent with the role of MdRunner. MMPropagator or MicrostatePropagator seem plausible, but perhaps too limited. Is there anything wrong with just "Propagator?" Examined another way, what is the data that is owned by objects of this type? And what is the fundamental change in program state before and after these objects have acted? I think it is pretty broad, and "gmx::Updater" does not seem inappropriate as a base class name.

#4 - 05/08/2019 05:21 PM - Berk Hess

I confused the mdp option names with the code object names. Ignore my comments above.

#5 - 05/08/2019 05:56 PM - Mark Abraham

The brains trust met in the dev telco today. The high level thing we will call Simulator, the microstate level thing we will call Integrator.

We need to change the .mdp option called integrator, but that's a trickier question. Someone needs to have a concrete proposal or two to debate.

#6 - 05/23/2019 08:04 PM - Pascal Merz

re: mdp-option - how about simulation-algorithm?

#7 - 12/27/2019 04:24 PM - Paul Bauer

- Status changed from New to Resolved

this has been resolved already

#8 - 12/27/2019 04:25 PM - Paul Bauer

- Status changed from Resolved to Closed