GROMACS - Feature #2581

User interface for hybrid Monte Carlo

07/20/2018 03:41 PM - Sebastian Wingbermühle

Status:	Accepted	
Priority:	Normal	
Assignee:		
Category:	mdrun	
Target version:	2020	
Difficulty:	uncategorized	

Description

This is about the user interface for a hybrid MC/MD integrator in GROMACS 2019 (or later).
The hybrid MC/MD integrator discussed here was first proposed by Mehlig et al. (1992) and includes the following steps:
1.) Draw random initial velocities from a Maxwell-Boltzmann distribution.
2.) "Standard" MD propagation in the NVE ensemble.
3.) A Metropolis step accepting or rejecting the resulting configuration based on the change in total energy: min(1, exp{-beta[E_tot (x_new) - E_tot (x_init)]}). In case of rejection, we rewind to the last accepted configuration.
NOTE: If you use different values of beta for the kinetic and the potential energy, you can draw the initial velocities at a temperature that is different from the temperature of the ensemble you wish to sample.

The implementation consists of three classes (HybridMCMDVelocities (step 1), MetropolisStepMehlig (step 3), AcceptOrRewind (step 3)). HybridMCMDVelocities and AcceptOrRewind are integrated in the implementation of md-vv in md.cpp and interact with MetropolisStepMehlig when needed. Thus, the implementation of the hybrid MC/MD integrator adds a Metropolis step to the (already implemented) velocity-Verlet integrator. Therefore, the user interface for the hybrid MC/MD integrator will provide a set of parameters to control the Metropolis step; all parameters affecting the velocity Verlet integrator have to be specified separately and like for a "standard" MD simulation.
Currently, the user interface contains five parameters:

```
ir->bDoHybridMCMD = get_eeenum(&inp, "hmc", yesno_names, wi);
hmc->nstHybridMCMD           = get_eint(inp, "nsthmc", -1, wi);
hmc->seedHybridMCMD          = get_eint(inp, "hmc-seed", -1, wi);
hmc->tempEnsembleHybridMCMD  = get_ereal(inp, "hmc-ens-temp", -1, wi);
hmc->tempVelocitiesHybridMCMD = get_ereal(inp, "hmc-vel-temp", -1, wi);
```

Name (for user)	Type	Name (internal)	Explanation
hmc	bool (yes/no)	bDoHybridMCMD;	Whether to use hybrid MC/MD or not
nsthmc	int	nstHybridMCMD;	The frequency at which the Metropolis criterion is evaluated
hmc-seed	int	seedHybridMCMD;	The random seed for the Metropolis step
hmc-ens-temp	real	tempEnsembleHybridMCMD;	The ensemble temperature used together with the potential energy in the Metropolis criterion
hmc-vel-temp	real	tempVelocitiesHybridMCMD;	The temperature used to generate initial velocities (can be different from the ensemble temperature)

Default values:
- nstHybridMCMD = 0 means "no hybrid MC/MD"
- seedHybridMCMD = ~~-1 means "randomly choose a new random seed every run"~~
tempEnsembleHybridMCMD has to be specified, no default value
- tempVelocitiesHybridMCMD = tempEnsembleHybridMCMD if user does not specify

I suggest to enforce the following dependencies between user input parameters (plus justification in brackets):
- if hybrid MC/MD, enforce md-vv (nothing else is implemented)
- nstHybridMCMD = nstcalcenergy (we need up-to-date energies when evaluating the Metropolis criterion, but it is not necessary to communicate energies during the NVE run)
- nstxout, nstxout_compressed, nstlog (maybe), nstenergy (maybe) have to be a multiple of nstHybridMCMD (make sure that users only use data from "Metropolised" configurations)
- if hybrid MC/MD, users will not be allowed to use temperature coupling options (the hybrid MC/MD uses NVE simulations to propose new configurations and the ensemble temperature is determined by the value used in the Metropolis criterion)
- if hybrid MC/MD, turn off pme-tuning and dynamic loadbalancing (will be implemented later)

Moreover, there has to be at least a warning that the hybrid MC/MD integrator will only yield correct output data reliably if combined with pure MD integration, no kind of enhanced sampling (umbrella sampling/pull-code, replica exchange, expanded ensemble, ...). Quantities that require moving averages (e.g. distance or orientation restraints) are an unsolved problem - especially if the moving average has to be updated very frequently, i.e. more frequently than the user wishes to apply the Metropolis criterion. Any

suggestions how to deal with them?

What do you think about this user interface? Any comments/ideas/suggestions/questions are highly welcome!

History

#1 - 10/15/2018 05:21 PM - Mark Abraham

- Status changed from New to Accepted

- Target version changed from 2019 to 2020

Out of time for 2019, but we should work on this in future!