Hamiltonian replica exchange for LAMMPS

Robert Darkins
University College London

July 17, 2023

Contents

1 Introduction

This LAMMPS package implements an `hrex` command and numerous fixes to perform Hamiltonian replica exchange (HREX). It also features a basic well-tempered metadynamics (WTMD) implementation with support for the hybrid WTMD-HREX method.

2 Theory

2.1 HREX

The purpose of HREX is to improve the ergodicity of a molecular simulation by allowing its configuration to randomly walk through a space of different ensembles (replicas) where the barriers that impede sampling are lowered through modified Hamiltonians.

HREX consists of N_R replicas running concurrently. Replica i has the Hamiltonian

$$\mathcal{H}_i = \mathcal{H} + \lambda_i \Delta U \tag{1}$$

where $\mathcal{H}$ is the Hamiltonian being investigated, the alchemical parameters λ_i increase monotonically, typically from $\lambda_1 = 0$ to $\lambda_{N_R} = 1$, and ΔU is a modification to the potential energy surface intended to reduce sampling barriers.

The atomic configuration x_i of replica i is periodically swapped with its neighbouring replica x_{i+1} with probability

$$P_i = \min\left(1, \frac{e^{-\beta E_i(x_{(i+1)})} e^{-\beta E_{i+1}(x_i)}}{e^{-\beta E_i(x_i)} e^{-\beta E_{i+1}(x_{(i+1)})}}\right) \tag{2}$$

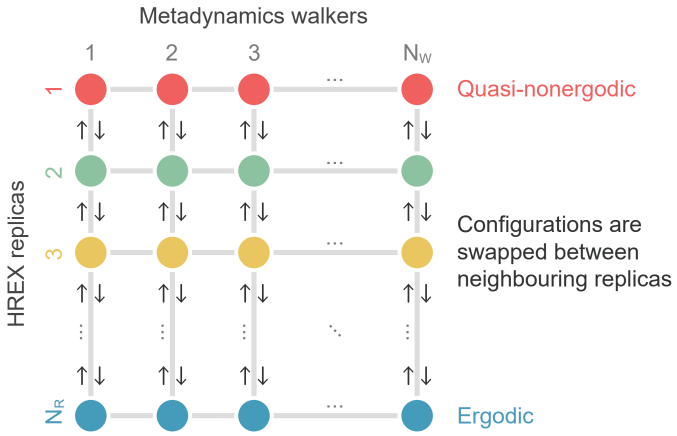

Figure 1: WTMD-HREX comprises $N_R \times N_W$ simulations (circles). Each walker randomly swaps atomic configurations between neighbouring replicas according to the Metropolis criterion.

where $E_i = \lambda_i \Delta U$. When i is even (odd), swapping attempts are made between replicas i and $i+1$ at times corresponding to even (odd) multiples of τ_{swap}. The first and last replicas therefore attempt swaps at intervals of $2\tau_{\text{swap}}$.

2.2 WTMD-HREX

The purpose of the hybrid WTMD-HREX method is to use WTMD to compute the free energy surface $F(\xi)$ as a function of the collective variables (CVs) ξ while accelerating the sampling of slow coordinates orthogonal to the CVs using HREX.

A total of $N_R \times N_W$ simulations are run concurrently, where N_R is the number of replicas and N_W is the number of metadynamics walkers per replica (Figure 1). Each simulation (i, j) is denoted by its replica i and walker j.

The N_W walkers assigned to replica i share the time-dependent Hamiltonian

$$\mathcal{H}_i = \mathcal{H} + \lambda_i \Delta U + V_i \tag{3}$$

where the alchemical parameters λ_i increase monotonically from $\lambda_1 = 0$ to $\lambda_{N_R} = 1$, ΔU is a modification to the potential energy surface, and V_i is the well-tempered multiple-walkers metadynamics bias potential which is built up iteratively at time intervals of τ_{deposit}. On the nth iteration,

$$V_{i,n}(\xi) = \sum_{j}^{N_W} \sum_{k}^{n} K(\xi, \xi_{ij,k}) \exp\left(-\beta V_{i,k-1}(\xi_{ij,k})/(\gamma_i - 1)\right), \tag{4}$$

where $\beta = (k_B T)^{-1}$, k_B is the Boltzmann constant, T is the temperature, $\gamma_i > 1$ are the bias factors, $\xi_{ij,k}$ are the CVs of simulation (i, j) at iteration k, and the Gaussian kernel is

$$K(\xi, \xi') = h \exp\left(-\frac{1}{2}(\xi - \xi')^T \Sigma^{-1} (\xi - \xi')\right) \tag{5}$$

where h is the initial Gaussian height and $\Sigma_{ij} = \sigma_i^2 \delta_{ij}$ is a diagonal matrix of variances.

For each simulation (i, j), the atomic configuration x_{ij} is periodically swapped with its neighbouring replica $x_{(i+1)j}$ with probability

$$P_{ij} = \min\left(1, \frac{e^{-\beta E_i(x_{(i+1)j})} e^{-\beta E_{i+1}(x_{ij})}}{e^{-\beta E_i(x_{ij})} e^{-\beta E_{i+1}(x_{(i+1)j})}}\right) \tag{6}$$

where $E_i = \lambda_i \Delta U + V_i$. When i is even (odd), swapping attempts are made between replicas i and $i+1$ at times corresponding to even (odd) multiples of τ_{swap}.

Finally, the free energy surface for replica i is

$$F_i(\xi) = -\frac{\gamma_i}{\gamma_i - 1} \lim_{n \to \infty} V_{i,n}(\xi) \tag{7}$$

modulo a constant, and the free energy surface being sought is $F(\xi) = F_1(\xi)$.

3 Installation

Copy the contents of `hrex/` into the LAMMPS `src/` directory and build in the usual way. For more information on building LAMMPS and custom packages, consult the LAMMPS documentation. This package has been tested with LAMMPS version 3 Nov 2022.

4 Usage

This section briefly summarises how to use HREX and WTMD-HREX. A detailed command reference is provided in section 4.7.

4.1 Create partitions

To distribute $N_R \times N_W \times N_P$ processors across $N_R \times N_W$ simulations, where N_R is the number of replicas, N_W is the number of walkers per replica (which must equal one if HREX is being used without WTMD), and N_P is the number of processors per simulation, use the partition functionality of LAMMPS to create $N_R \times N_W$ partitions. For example, if $N_R = 5$, $N_W = 3$ and $N_P = 10$, then the processors would be partitioned at the command line using

```
$ mpirun -n 150 ${LAMMPS_EXE} -in inputfile -partition 15x10
```

Alternatively there is a `partition` command that can be invoked in the input file (consult the LAMMPS documentation). The number of replicas and walkers will be defined in the input file.

Note that although the theory section describes the exchange of configurations between replicas, each partition actually simulates a continuous trajectory and instead exchanges the Hamiltonians.

4.2 HREX

To perform an HREX simulation, use the `hrex` command provided by our package instead of the usual `run` command. Customisations such as the potential energy modification ΔU, the use of WTMD, and HREX dumps must all be specified using appropriate fixes before calling `hrex`, as detailed next.

4.3 Potential energy modification

All N_W partitions belonging to replica i will have $\lambda_i \Delta U$ added to their Hamiltonian. The values of λ_i are specified using the `hrex` command, but the modification ΔU must be specified using one or more of the `hrex/pot/*` fixes before calling `hrex`. We provide two such modifications:

- Fix `hrex/pot/srnb` reduces the short-range non-bonded interactions between all atoms in one group (e.g., a solute) with all atoms in another group (e.g., a solvent). It is used to weaken specific interactions.
- Fix `hrex/pot/hill` adds a Gaussian-shaped lump of energy to a particular location. It could be used to prevent certain atoms from visiting certain regions of space.

If multiple `hrex/pot/*` fixes are applied, they are combined additively. To create a custom potential energy modification, see Section 5.

4.4 WTMD-HREX

To use the hybrid WTMD-HREX method, WTMD must be enabled with an appropriate `hrex/metad/*` fix before calling `hrex`. We currently provide a single implementation of WTMD:

- Fix `hrex/metad/basic` supports well-tempering and multiple walkers. For CVs it supports the x and/or y and/or z position of the centre of mass of a group of atoms, and it also supports the distance r between the centres of mass of two groups of atoms.

Currently our package is not interfaced with any dedicated metadynamics packages such as PLUMED, although this would be possible, see Section 5

4.5 HREX dumps

Use `hrex/dump/*` fixes to output the progress of the HREX simulation, e.g., to record information about the configuration at each replica exchange. Currently a single dump is supported:

- Fix `hrex/dump/swap` creates a record of all swaps that occur between the replicas. It can restore an earlier state if the simulation is restarted.

To create a custom HREX dump, see Section 5.

4.6 Restarts

To restart an HREX or WTMD-HREX simulation, you are responsible for reloading the last atomic configuration for each partition (e.g., using `read_dump`, `read_data` or `read_restart`). The HREX and metadynamics features will automatically reload their previous states if their output files exist at the first invocation of `hrex`. Specifically, fix `hrex/dump/swap` will restore the last replica from its output file, and fix `hrex/metad/basic` will restore the bias potential of each replica.

4.7 Command reference

`hrex`

```
hrex N M fix-ID seed NR L1 ... LNR
```

- `N` = total number of timesteps to run (must be a multiple of `M`)
- `M` = attempt a swap every this many timesteps (`M` $= \tau_{\text{swap}}/\Delta t$)
- `fix-ID` = ID of the fix that controls the temperature
- `seed` = random seed for Metropolis exchange
- `NR` = number of replicas N_R
- `L1 ... LNR` = list of the alchemical parameters λ_1 ... λ_{N_R} for each replica

This command performs Hamiltonian replica exchange. The potential energy modification should be defined beforehand using one or more `hrex/pot/*` fixes. If an `hrex/metad/*` fix has been defined, then the hybrid WTMD-HREX method will be used. Multiple walkers will automatically be used if the number of partitions exceeds the number of replicas. The code determines the number of walkers by dividing the number of partitions by the number of replicas (the former must therefore be a multiple of the latter). If WTMD is not used, then the number of replicas must equal the number of partitions.

`hrex/pot/srnb`

```
fix ID group-ID hrex/pot/srnb group-ID2 scale_factor
```

- `ID` = fix ID
- `group-ID` = ID of group 1
- `group-ID2` = ID of group 2
- `scale_factor` = reduce the interaction energy by this fraction

This fix defines a potential energy modification of the form

$$\Delta U = -(\texttt{scale\_factor}) \sum_{ij} u_{ij} \tag{8}$$

where u_{ij} is the short-range non-bonded interaction energy between atom i and atom j, where i is summed over group 1 and j is summed over group 2. For example, if group 1 contains a single atom A and group 2 contains two atoms B and C, and the scale factor is 0.1, then the modification reduces the short-range non-bonded interaction energy between A-B and A-C by 10%.

hrex/pot/hill

```
fix ID group-ID hrex/pot/hill x y z h sigma
```

- `ID` = fix ID
- `group-ID` = ID of group to apply forces to
- `x, y, z` = position of energy penalty (length units)
- `h` = height of penalty (energy units)
- `sigma` = standard deviation of energy penalty (length units)

This fix adds a Gaussian-shaped lump of energy at position `(x,y,z)` of height `h` and width `sigma` to be experienced by the specified group of atoms. If `x`, `y` or `z` is `NULL`, then that dimension will be left out of the distance and force calculations.

hrex/metad/basic

```
fix ID group-ID hrex/metad/basic Q N h sigma T file keyword value ...
```

- `ID` = fix ID
- `group-ID` = ID of group 1
- `Q` = do not add to the bias potential until this many timesteps pass (convenient for relaxation)
- `N` = deposit a Gaussian to the bias potential every this many time steps
- `h` = initial height of Gaussians (energy units)
- `sigma` = standard deviation of Gaussians (length units)
- `T` = system temperature for well-tempering (temperature units)
- `file` = filename where a list of all deposited Gaussians are recorded
- One or more keyword/value pairs may be appended:
  - keyword = `x` or `y` or `z` or `r` or `bf`
  - `x` or `y` or `z` values = `min max skin k`
    - `min, max` = the CV (`x, y` or `z`) will not leave the interval `[min, max)`
    - `eps, k` = if the CV decreases below `min+eps` or increases above `max-eps`, then a harmonic potential with spring constant `k` will act to restore it
  - `r` values = `group-ID2 min max skin k`
    - `group-ID2` = ID of group 2
    - `min, max` = the CV `r` will not leave the interval `[min, max)`
    - `eps, k` = if the CV decreases below `min+eps` or increases above `max-eps`, then a harmonic potential with spring constant `k` will act to restore it
  - `bf` values = `bf1 ... bfNR`
    - `bfi` = the bias factor γ_i of replica `i`

This fix performs metadynamics. It supports the CVs `x` and/or `y` and/or `z` of the centre of mass of the group specified by `group-ID`, or the distance `r` between the centres of mass of `group-ID` and `group-ID2`. The distance `r` cannot be used in combination with the other CVs. A CV is only sampled if its keyword/value pair is specified (see `x, y, z` and `r`). All CVs have the same Gaussian width `sigma`.

The bias potential is stored on a grid where each CV spans its specified range `[min, max)` with a cell spacing of approximately `sigma/10`. If a CV leaves this range it will be wrapped back into the range using periodic boundaries. If you do not wish for this to happen, then you should specify a large enough constant k to keep the CVs away from their respective grid edges. The wall potential energy takes the form $\frac{1}{2}k\Delta^2$ where Δ is the distance the CV has penetrated the wall boundary at min+eps or max-eps.

To use well-tempering, specify the bias factors using the bf keyword. A bias factor of one (which is the default) disables well-tempering.

Every time a Gaussian is deposited it is recorded in a file named file.i where i is the replica ID. All N_W walkers assigned to replica i will write to the same file. The first line is a header

```
# sigma [x min max] [y min max] [z min max] [r min max]
```

and the subsequent lines take the form

```
[x] [y] [z] [r] h
```

where columns in square brackets are only included if the corresponding CV is used, and where h is the height of the deposited Gaussian multiplied by $\gamma_i/(\gamma_i - 1)$. This scaling factor corrects for well-tempering such that the free energy surface can be reconstructed by simply summing the Gaussians in the file. If the lines in the output file were divided into blocks of N_W lines, each block would contain the Gaussians deposited by each walker during the same time step, but the order of walkers within each block may vary.

If the files file.* already exist at the start of the simulation, the Gaussians will be restored from them for a seamless restart. Different bias factors may be used upon restarting.

Multiple walkers will automatically be used if the number of partitions exceeds the number of replicas, as described in the documentation for the hrex command.

hrex/dump/swap

```
fix ID group-ID hrex/dump/swap file
```

- ID = fix ID
- group-ID = ignored
- filename = output file

This creates a series of files named file.N where N is the partition ID. Each line in the output file records an event where the corresponding partition has swapped replicas. It has the format:

```
timestep i j lambda
```

where timestep is the timestep of the swap, i is the replica ID after the swap, j is the walker ID after the swap, and lambda is λ_i after the swap. The value of j will remain constant for each partition since there is no inter-walker swap, but it is included for completeness.

A line will be written at the first timestep to record the starting replica and walker of each partition. If the output files already exist when hrex is first called, then the last state of each file will be restored and subsequent swaps will be appended to the files.

5 Developing the code

The class hierarchy in our package is shown in Figure 2. The HREX class implements the HREX algorithm and is the core of the package. There are three base classes that define fixes (and which derive from a fourth base class FixHREX). These base classes have the following purposes:

- FixHREXPot = define potential energy modifications ΔU, add the auxiliary forces $-\nabla(\lambda_i \Delta U)$.
- FixHREXMetad = implement metadynamics, add the forces $-\nabla V_i$.
- FixHREXDump = record the simulation progress.

The HREX class invokes them as follows:

- HREX requests from each instance of FixHREXPot its contribution to ΔU for the Metropolis exchange.

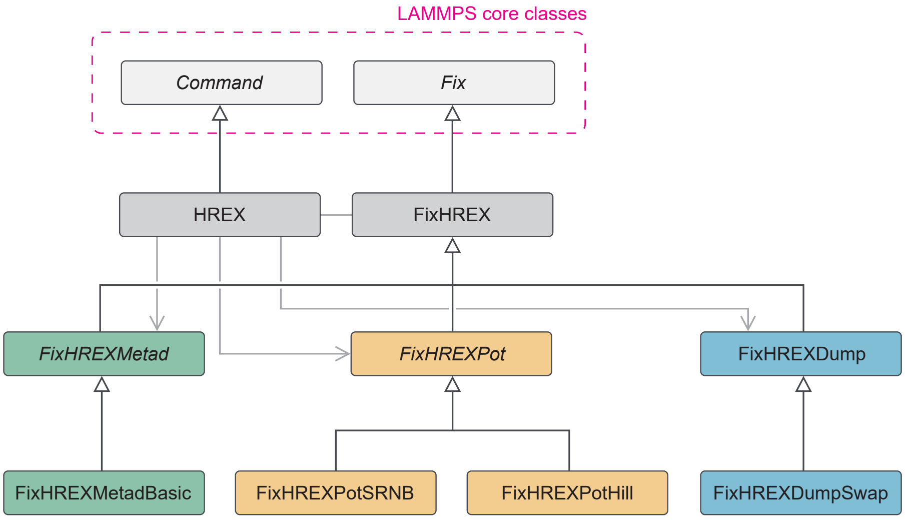

Figure 2: UML diagram of the class hierarchy in the HREX package.

- `HREX` requests from each instance of `FixHREXMetad` the current CV and the value of the bias potential for the Metropolis exchange. The class `HREX` is responsible for exchanging the bias potentials between replicas in the event of a replica exchange.
- `HREX` notifies all three classes through functions inherited from `FixHREX` that an HREX simulation has begun, ended, and that a replica exchange has taken place.

Two partitions exchange Hamiltonians by simply exchanging replica IDs and, if using metadynamics, exchanging metadynamics bias potentials. This latter operation requires that the metadynamics implementation provide routines for packing and unpacking the bias potential. Note that multiple walkers is implemented in the `FixHREXMetad` base class.

New functionality can be added by creating a new class that derives from the appropriate base class, `FixHREXPot`, `FixHREXMetad` or `FixHREXDump`. We suggest reading the code comments in the base classes and examining the existing derived classes for illustrative examples.

Contributing

If you wish to contribute your changes to the package, please submit a pull request to the GitHub repository for review.

6 Legal

This software is freely available open-source code, distributed under the terms of the GNU General Public License Version 2 (GPLv2). This software is provided “as is” without warranty of any kind, either express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, or non-infringement. Use of this software is at your own risk. The author of this code cannot be held liable for any damages arising from the use of this software.