FHI-aims Cheat Sheet: Replica Exchange Molecular Dynamics

FHI-aims team

August 7, 2013

Replica-exchange molecular dynamics (REMD) with FHI-aims makes use of the SGE batch-queuing system. Some precautions have to be made, but are explained in the following. **Please make sure to read the FHI-aims-Manual (chapter 4.10 - Script based parallel tempering).**

How to begin?

The following files must be present in the working directory:

- `control.in.basic`
  - The "usual" `control.in` file. Please make sure that the `MD_time_step` is set. Other variables concerning geometry relaxation and MD will be ignored.
- `geometry.in.basic`
  - The "usual" `geometry.in` file. Every replica will start from this geometry. It is possible to provide different geometries for different replicas by making use of the optional `list_of_geometries` file (please refer to the `FHI-aims`-Manual).
- `control.in.rex`
  - This file provides REMD parameters, for example:

```
n_rex   5
temps   300.00 350.00 400.00 450.00 500.00
freq            0.04 # 1/ps
MAX_steps       15
```

  - In this example, 5 replicas with temperatures from 300 to 500 K are requested. Temperature swaps are attempted every 0.04 ps. The maximum number of replica-exchange temperature swap attempts is set to 15, thus the whole simulation will end after 0.6 ps ($= 15 \cdot 0.04$ ps).
- `rex.AIMS.pl`
  - Managing perl script. Normally this file does not need to be altered by the user. Some statistics can be provided (also on-the-fly) by running

```
perl rex.AIMS.pl stat log_rex
```

- `submit.rex`

- ▶ The batch script. Needs to be altered by the user according to his needs. In particular:
  - ∗ select the total number of slots in line

    ```
    #$ -pe impi <#slots>
    ```

  - ∗ set the location of the aims binary in line

    ```
    binary='<path-to-aims-binary>'
    ```

  - ∗ set `type` variable to `'init'` or `'restart'`, depending on whether the run should be started from scratch or restarted
  - ∗ set the number of slots per node (host) in line

    ```
    ncpupn=<#SlotsPerNode>
    ```

  - ∗ in order to distribute the slots accordingly, the machine file is read in and its location might need to be adjusted for the specific cluster system in use, in particular:

    ```
    cat $TMPDIR/machines > $hostfile
    ```

- The files `rex.AIMS.pl` and `submit.rex` are provided with the distribution and are contained in the subdirectory `utilities/REX`.

Running REMD and output

- The job is submitted via

  ```
  qsub submit.rex
  ```

  or, if one wants to run REMD locally, type

  ```
  bash run_rex.sh
  ```

- ▶ `run_rex.sh` serves as a substitution for `submit.rex` if the SGE is not available. In any case, it is a better idea to use `submit.rex` (if possible) because of the more sophisticated distribution of jobs over the available slots (CPU).

- The script then automatically does the following:
  - For each replica a directory `rex_??` is created, the required files are copied into those directories and the MD runs with `FHI-aims` are applied.
  - The replica exchange temperature swaps are managed automatically.
  - Output is printed.

- In each directory `rex_??`, some interesting output can be found:
  - `temp.out`
    - ∗ The regular `FHI-aims` output for the parallel tempering step.
    - ∗ After each parallel tempering step, the file will be appended to `out.????` (in the working directory), corresponding to the particular replica temperature.

    - That means `out.????` contains the full `FHI-aims` output for a given temperature.
  - `energy.trajectory`
    - Cumulative (i.e. appended after each attempted swap) energy trajectory for the replica.
  - `out.xyz`
    - Cumulative geometry trajectory, in `xyz` format

- `log_rex` in the working directory
  - This file contains useful information on the swapping process.
  - Some statistics can be provided (also on-the-fly) by running

    ```
    perl rex.AIMS.pl stat log_rex
    ```

  - The file also contains some information (and warnings) concerning the selected number of slots.

Restarting a parallel tempering run

Restart a time-exceeded run

When a job exceeds its time limit and is killed in the process, a message is written to `log_rex`:

```
  WARNING: rex_??/temp.out Not converged?
Please check this problem before continuing.
```

In that case, it is possible to restart the run by setting

```
type='restart'
```

in `submit.rex`.

Restart an already completed run

When a job has completed, i.e. the desired number of Replica Exchange steps has been performed, it is possible to restart (prolong) the run by doing the following:

- in `submit.rex`, set

  ```
  type='restart'
  ```

- in `control.in.rex`, set

  ```
  MAX_steps        <#MaxSteps>
  ```

  according to the (new) desired maximum number of steps.

- replace the third number in `rex_par` with that same number `<#MaxSteps>`

Attention!

These cheat sheets are not intended to substitute for reading the manuals of the programs involved and the original literature cited therein!
Have fun!