# Using a Neural Network Potential Embedding in a MM Model using NAMD

Shae-Lynn Lahey, Từ Nguyễn Thiên Phúc, and Christopher N. Rowley

The NNP/MM embeds a Neural Network Potential into a conventional molecular mechanical (MM) model. We have implemented this using the Custom QM/MM features of NAMD 2.13, which interface NAMD with the TorchANI NNP python library.

The ani-server.py script must be executed and continue to run in the background before NAMD is executed. This server interfaces with the TorchANI library. The ani-client.py script is executed by NAMD at each energy/gradient step, which communicates with the ani-server.py. NAMD writes the atomic coordinates to disk. The ani-client.py script polls the ani-server.py. The socket communication is performed through a file handle (/tmp/ani_socket) To reduce the rate of these file operations, we use a local RAM to store these files, although, in practice, this does not change the speed of the simulations significantly.

## Installation

1. Install TorchANI.
2. Install NAMD. Version 2.13 or later is required. The single-node version (e.g., Linux-x86_64-multicore) is fine for this because the ANI calculations will be slower than the MM component, so MPI parallelization.
3. Install client.py and server.py scripts into an accessible location.

## Execution

An example SLURM submission script for a NAMD/MM simulation

```
export RUNDIR=/dev/shm/$SLURM_JOB_ID/
mkdir $RUNDIR

module load nixpkgs/16.09  intel/2016.4
module load namd-multicore/2.13

module load python/3.7.0

python3 server.py >& server.\$SLURM_JOB_ID.log&
sleep 10 # wait to make sure server has initialized first
namd2  +p4 namd_input.conf > namd_output
```

The custom QM features of NAMD have to be activated in the NAMD configuration file:

```
set rundir $env(RUNDIR)
qmforces on
qmParamPDB qmmm.pdb
qmSoftware custom
qmexecpath client.py
qmBaseDir  $rundir
QMColumn occ
qmChargeMode none
qmElecEmbed off
```

The occupancy column of the qmmm.pdb specifies which atoms should be treated using the NNP.

## Examples

The examples directory of the GitHub repository contains the input files an NNP/MM simulation of erlotinib in liquid water, where erlotinib is represented using the NNP and the water molecules are calculated using the TIP3P MM model.

## Practical Notes

Note that the AN-1/ANI-1ccX model should not be used for compounds with charged functional groups. Only molecules comprised of C, N, O, and H atoms are supported by the ANI-1x/ANI-1ccX.

The atoms in the NNP region must not cross the boundary of the simulation cell. The simplest way to do this is to translate the NNP system to the origin (0, 0, 0). Alternatively, the CellOrigin NAMD keyword can be used to place the center of the simulation cell at the center of mass of the NNP region. In each case, it is advisable to restrain the NNP atoms with a harmonic potential so that they do not diffusion across the boundary during the simulation.

```
cellbasisvector1 39.4435539101   0.0  0.0
cellbasisvector2 0.0 38.2819366381 0.0
cellbasisvector3 0.0 0.0 38.9952104018
cellorigin 21.31 0.50 52.4
```

## Citing

Researchers using this code should cite the following paper:

Lahey S.-L. J., Rowley, C. N., Simulating Protein-Ligand Binding with Neural Network Potentials, *Chemical Science,* **2020**, doi: **10.1039/C9SC06017K**

```
@Article{NNP_MM_2020,
author ="Lahey, Shae-Lynn J and Rowley, Christopher N.",
title  ="Simulating Protein-Ligand Binding with Neural Network Potentials",
journal  ="Chem. Sci.",
year  ="2020",
pages  ="-",
publisher  ="The Royal Society of Chemistry",
doi  ="10.1039/C9SC06017K",
url  ="http://dx.doi.org/10.1039/C9SC06017K",
}
```