NAME

OpenMMPerformMDSimulation.py - Perform a MD simulation

SYNOPSIS

OpenMMPerformMDSimulation.py [--forcefieldParams <Name,Value,..>] [--freezeAtoms <yes or no>] [--freezeAtomsParams <Name,Value,..>] [--integratorParams <Name,Value,..>] [--mode <NVT or NPT>] [--outputParams <Name,Value,..>] [--outfileDir <outfiledir>] [--outfilePrefix <text>] [--outPlotParams <Name,Value,...>] [--overwrite] [--platform <text>] [--platformParams <Name,Value,..>] [--restart <yes or no>] [--restartParams <Name,Value,..>] [--restraintAtoms <yes or no>] [--restraintAtomsParams <Name,Value,..>] [--restraintSpringConstant <number>] [--simulationParams <Name,Value,..>] [--smallMolFile <SmallMolFile>] [--smallMolID <text>] [--systemParams <Name,Value,..>] [--waterBox <yes or no>] [--waterBoxParams <Name,Value,..>] [-w <dir>] -i <infile>

OpenMMPerformMDSimulation.py -h | --help | -e | --examples

DESCRIPTION

Perform a MD simulation using an NPT or NVT statistical ensemble. You may run a simulation using a macromolecule or a macromolecule in a complex with small molecule. By default, the system is minimized and equilibrated before the production run.

The input file must contain a macromolecule already prepared for simulation. The preparation of the macromolecule for a simulation generally involves the following: identification and replacement non-standard residues; addition of missing residues; addition of missing heavy atoms; addition of missing hydrogens; addition of a water box which is optional.

In addition, the small molecule input file must contain a molecule already prepared for simulation. It must contain appropriate 3D coordinates relative to the macromolecule along with no missing hydrogens.

You may optionally add a water box and freeze/restraint atoms for the simulation.

The supported macromolecule input file formats are: PDB (.pdb) and CIF (.cif)

The supported small molecule input file format are : SD (.sdf, .sd)

Possible outfile prefixes:

```
<InfileRoot>_<Mode>
<InfileRoot>_Solvated_<Mode>
<InfileRoot>_<SmallMolFileRoot>_Complex_<Mode>,
<InfileRoot>_<SmallMolFileRoot>_Complex_Solvated_<Mode>
```

Possible output files:

```
<OutfilePrefix>.<pdb or cif> [ Initial sytem ]
<OutfilePrefix>.<dcd or xtc>

<OutfilePrefix>_Reimaged.<pdb or cif> [ First frame ]
<OutfilePrefix>_Reimaged.<dcd or xtc>

<OutfilePrefix>_Minimized.<pdb or cif>
<OutfilePrefix>_Equilibrated.<pdb or cif>
<OutfilePrefix>_Final.<pdb or cif> [ Final system ]

<OutfilePrefix>.chk
<OutfilePrefix>.csv
<OutfilePrefix>_Minimization.csv
<OutfilePrefix>_FinalState.chk
<OutfilePrefix>_FinalState.xml

<OutfilePrefix>_System.xml
<OutfilePrefix>_Integrator.xml

<OutfilePrefix>_<DataOutTypePlotY1>Plot.<outExt>
<OutfilePrefix>_<DataOutTypePlotY2>Plot.<outExt>
... ... ...
```

The reimaged PDB file, <OutfilePrefix>_Reimaged.pdb, corresponds to the first frame in the trajectory. The reimaged trajectory file contains all the frames aligned to the first frame after reimaging of the frames for periodic systems.

OPTIONS

-e, --examples

Print examples.

-f, --forcefieldParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs for biopolymer, water, and small molecule forcefields.

The supported parameter names along with their default values are are shown below:

```
biopolymer, amber14-all.xml  [ Possible values: Any Valid value ]
smallMolecule, openff-2.2.1  [ Possible values: Any Valid value ]
water, auto  [ Possible values: Any Valid value ]
additional, none [ Possible values: Space delimited list of any
    valid value ]
```

Possible biopolymer forcefield values:

```
amber14-all.xml, amber99sb.xml, amber99sbildn.xml, amber03.xml,
amber10.xml
charmm36.xml, charmm_polar_2019.xml
amoeba2018.xml
```

Possible small molecule forcefield values:

```
openff-2.2.1, openff-2.0.0, openff-1.3.1, openff-1.2.1,
openff-1.1.1, openff-1.1.0,...
smirnoff99Frosst-1.1.0, smirnoff99Frosst-1.0.9,...
gaff-2.11, gaff-2.1, gaff-1.81, gaff-1.8, gaff-1.4,...
```

The default water forcefield valus is dependent on the type of the biopolymer forcefield as shown below:

```
Amber: amber14/tip3pfb.xml
CHARMM: charmm36/water.xml or None for charmm_polar_2019.xml
Amoeba: None (Explicit)
```

Possible water forcefield values:

```
amber14/tip3p.xml, amber14/tip3pfb.xml, amber14/spce.xml,
amber14/tip4pew.xml, amber14/tip4pfb.xml,
charmm36/water.xml, charmm36/tip3p-pme-b.xml,
charmm36/tip3p-pme-f.xml, charmm36/spce.xml,
charmm36/tip4pew.xml, charmm36/tip4p2005.xml,
charmm36/tip5p.xml, charmm36/tip5pew.xml,
implicit/obc2.xml, implicit/GBn.xml, implicit/GBn2.xml,
amoeba2018_gk.xml (Implict water)
None (Explicit water for amoeba)
```

The additional forcefield value is a space delimited list of any valid forcefield values and is passed on to the OpenMMForcefields SystemGenerator along with the specified forcefield values for biopolymer, water, and mall molecule. Possible additional forcefield values are:

```
amber14/DNA.OL15.xml amber14/RNA.OL3.xml
amber14/lipid17.xml amber14/GLYCAM_06j-1.xml
... ... ...
```

You may specify any valid forcefield names supported by OpenMM. No explicit validation is performed.

--freezeAtoms <yes or no> [default: no]

Freeze atoms during a simulation. The specified atoms are kept completely fixed by setting their masses to zero. Their positions do not change during local energy minimization and MD simulation, and they do not contribute to the kinetic energy of the system.

--freezeAtomsParams <Name,Value,..>

A comma delimited list of parameter name and value pairs for freezing atoms during a simulation. You must specify these parameters for 'yes' value of '--freezeAtoms' option.

The supported parameter names along with their default values are are shown below:

```
selection, none [ Possible values: CAlphaProtein, Ions, Ligand,
    Protein, Residues, or Water ]
selectionSpec, auto [ Possible values: A space delimited list of
    residue names ]
negate, no [ Possible values: yes or no ]
```

A brief description of parameters is provided below:

```
selection: Atom selection to freeze.
selectionSpec: A space delimited list of residue names for
    selecting atoms to freeze. You must specify its value during
    'Ligand' and 'Protein' value for 'selection'. The default values
    are automatically set for 'CAlphaProtein', 'Ions', 'Protein',
    and 'Water' values of 'selection' as shown below:

    CAlphaProtein: List of stadard protein residues from pdbfixer
        for selecting CAlpha atoms.
    Ions: Li Na K Rb Cs Cl Br F I
    Water: HOH
    Protein: List of standard protein residues from pdbfixer.

negate: Negate atom selection match to select atoms for freezing.
```

In addition, you may specify an explicit space delimited list of residue names using 'selectionSpec' for any 'selection". The specified residue names are appended to the appropriate default values during the selection of atoms for freezing.

-h, --help

Print this help message.

-i, --infile <infile>

Input file name containing a macromolecule.

--integratorParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs for integrator during a simulation.

The supported parameter names along with their default values are are shown below:

```
integrator, LangevinMiddle [ Possible values: LangevinMiddle,
    Langevin, NoseHoover, Brownian ]

randomSeed, auto [ Possible values: > 0 ]

frictionCoefficient, 1.0 [ Units: 1/ps ]
stepSize, auto [ Units: fs; Default value: 4 fs during yes value of
    hydrogen mass repartioning with no freezing/restraining of atoms;
    otherwsie, 2 fs ]
temperature, 300.0 [ Units: kelvin ]

barostat, MonteCarlo [ Possible values: MonteCarlo or
    MonteCarloMembrane ]
barostatInterval, 25
pressure, 1.0 [ Units: atm ]

Parameters used only for MonteCarloMembraneBarostat with default
values corresponding to Amber forcefields:

surfaceTension, 0.0 [ Units: atm*A. It is automatically converted
    into OpenMM default units of atm*nm before its usage.  ]
xymode,  Isotropic [ Possible values: Anisotropic or  Isotropic ]
zmode,  Free [ Possible values: Free or  Fixed ]
```

A brief description of parameters is provided below:

```
integrator: Type of integrator

randomSeed: Random number seed for barostat and integrator. Not
    supported for NoseHoover integrator.
```

```
            frictionCoefficient: Friction coefficient for coupling the system to
                the heat bath..
            stepSize: Simulation time step size.
            temperature: Simulation temperature.

            barostat: Barostat type.
            barostatInterval: Barostat interval step size, in terms of time
                step size, for applying Monte Carlo pressure changes during
                NPT simulation.
            pressure: Pressure during NPT simulation.

            Parameters used only for MonteCarloMembraneBarostat:

            surfaceTension: Surface tension acting on the system.
            xymode: Behavior along X and Y axes. You may allow the X and Y axes
                to vary independently of each other or always scale them by the same
                amount to keep the ratio of their lengths constant.
            zmode: Beahvior along Z axis. You may allow the Z axis to vary
                independently of the other axes or keep it fixed.
```

-m, --mode <NPT or NVT> [default: NPT]

Type of statistical ensemble to use for simulation. Possible values: NPT (constant Number of particles, Pressure, and Temperature) or NVT ((constant Number of particles, Volume and Temperature)

-o, --outfileDir <outfiledir>

Output files directory. Default: Current working directory.

--outfilePrefix <text> [default: auto]

File prefix for generating the names of output files. The default value depends on the names of input files for macromolecule and small molecule along with the type of statistical ensemble and the nature of the solvation.

The possible values for outfile prefix are shown below:

```
<InfileRoot>_<Mode>
<InfileRoot>_Solvated_<Mode>
<InfileRoot>_<SmallMolFileRoot>_Complex_<Mode>,
<InfileRoot>_<SmallMolFileRoot>_Complex_Solvated_<Mode>
```

--outputParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs for generating output during a simulation.

The supported parameter names along with their default values are are shown below:

```
checkpoint, no  [ Possible values: yes or no ]
checkpointFile, auto  [ Default: <OutfilePrefix>.chk ]
checkpointSteps, 10000

dataOutType, auto [ Possible values: A space delimited list of valid
    parameter names.
    NPT simulation default: Density Step Speed Progress
    PotentialEnergy Temperature Time.
    NVT simulation default: Step Speed Progress PotentialEnergy
    Temperature Time Volume
    Other valid names: ElapsedTime RemainingTime KineticEnergy
    TotalEnergy  ]

dataLog, yes  [ Possible values: yes or no ]
dataLogFile, auto  [ Default: <OutfilePrefix>.csv ]
dataLogSteps, 1000

dataStdout, no  [ Possible values: yes or no ]
dataStdoutSteps, 1000

dataOutTypePlot, yes  [ Possible values: yes or no ]
dataOutTypePlotX, auto  [ Default: Time; Possible values: Step or
    Time ]
dataOutTypePlotY, auto  [ Possible values: A space delimited list
    of valid parameter names specified for dataOutType.
```

```
        NPT simulation default: Density PotentialEnergy Temperature
        NVT simulation default: PotentialEnergy Temperature Volume
        Other valid names: KineticEnergy TotalEnergy]

    minimizationDataSteps, 100
    minimizationDataStdout, no  [ Possible values: yes or no ]
    minimizationDataLog, no  [ Possible values: yes or no ]
    minimizationDataLogFile, auto  [ Default:
        <OutfilePrefix>_MinimizationOut.csv ]
    minimizationDataOutType, auto [ Possible values: A space delimited
        list of valid parameter names.  Default: SystemEnergy
        RestraintEnergy MaxConstraintError.
        Other valid names: RestraintStrength ]

    pdbOutFormat, PDB  [ Possible values: PDB or CIF ]
    pdbOutKeepIDs, yes  [ Possible values: yes or no ]

    pdbOutMinimized, no  [ Possible values: yes or no ]
    pdbOutEquilibrated, no  [ Possible values: yes or no ]
    pdbOutFinal, no  [ Possible values: yes or no ]

    saveFinalStateCheckpoint, yes  [ Possible values: yes or no ]
    saveFinalStateCheckpointFile, auto  [ Default:
        <OutfilePrefix>_FinalState.chk ]
    saveFinalStateXML, no  [ Possible values: yes or no ]
    saveFinalStateXMLFile, auto  [ Default:
        <OutfilePrefix>_FinalState.xml]

    traj, yes  [ Possible values: yes or no ]
    trajFile, auto  [ Default: <OutfilePrefix>.<TrajFormat> ]
    trajFormat, DCD  [ Possible values: DCD or XTC ]
    trajSteps, 10000 [ The default value corresponds to 40 ps for step
        size of 4 fs. ]

    xmlSystemOut, no  [ Possible values: yes or no ]
    xmlSystemFile, auto  [ Default: <OutfilePrefix>_System.xml ]
    xmlIntegratorOut, no  [ Possible values: yes or no ]
    xmlIntegratorFile, auto  [ Default: <OutfilePrefix>_Integrator.xml ]
```

A brief description of parameters is provided below:

```
    checkpoint: Write intermediate checkpoint file.
    checkpointFile: Intermediate checkpoint file name.
    checkpointSteps: Frequency of writing intermediate checkpoint file.

    dataOutType: Type of data to write to stdout and log file.

    dataLog: Write data to log file.
    dataLogFile: Data log file name.
    dataLogSteps: Frequency of writing data to log file.

    dataStdout: Write data to stdout.
    dataStdoutSteps: Frequency of writing data to stdout.

    dataOutTypePlot: Generate plots using data written to log file.
    dataOutTypePlotX: Data out type to plot on X axis.
    dataOutTypePlotY: Data out types to plot on Y axis. An individual plot
        is generated for each pair of X and Y vaues to be plotted.

    minimizationDataSteps: Frequency of writing data to stdout
        and log file.
    minimizationDataStdout: Write data to stdout.
    minimizationDataLog: Write data to log file.
    minimizationDataLogFile: Data log fie name.
    minimizationDataOutType: Type of data to write to stdout
        and log file.

    saveFinalStateCheckpoint: Save final state checkpoint file.
    saveFinalStateCheckpointFile: Name of final state checkpoint file.
```

```
saveFinalStateXML: Save final state XML file.
saveFinalStateXMLFile: Name of final state XML file.

pdbOutFormat: Format of output PDB files.
pdbOutKeepIDs: Keep existing chain and residue IDs.

pdbOutMinimized: Write PDB file after minimization.
pdbOutEquilibrated: Write PDB file after equilibration.
pdbOutFinal: Write final PDB file after production run.

traj: Write out trajectory file.
trajFile: Trajectory file name.
trajFormat: Trajectory file format.
trajSteps: Frequency of writing trajectory file.

xmlSystemOut: Write system XML file.
xmlSystemFile: System XML file name.
xmlIntegratorOut: Write integrator XML file.
xmlIntegratorFile: Integrator XML file name.
```

--outPlotParams <Name,Value,...> [default: auto]

A comma delimited list of parameter name and value pairs for generating plots using Seaborn module. The supported parameter names along with their default values are shown below:

```
type,linepoint,outExt,svg,width,10,height,5.6,
titleWeight,bold,labelWeight,bold, style,darkgrid,
palette,deep,font,sans-serif,fontScale,1,
context,notebook
```

Possible values:

```
type: linepoint, scatter, or line. Both points and lines are drawn
    for linepoint plot type.
outExt: Any valid format supported by Python module Matplotlib.
    For example: PDF (.pdf), PNG (.png), PS (.ps), SVG (.svg)
titleWeight, labelWeight: Font weight for title and axes labels.
    Any valid value.
style: darkgrid, whitegrid, dark, white, ticks
palette: deep, muted, pastel, dark, bright, colorblind
font: Any valid font name
context: paper, notebook, talk, poster, or any valid name
```

--overwrite

Overwrite existing files.

-p, --platform <text> [default: CPU]

Platform to use for running MD simulation. Possible values: CPU, CUDA, OpenCL, or Reference.

--platformParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs to configure platform for running MD simulation.

The supported parameter names along with their default values for different platforms are shown below:

```
CPU:

threads, 1  [ Possible value: >= 0 or auto.  The value of 'auto'
    or zero implies the use of all available CPUs for threading. ]

CUDA:

deviceIndex, auto  [ Possible values: 0, '0 1' etc. ]
deterministicForces, auto [ Possible values: yes or no ]
precision, single  [ Possible values: single, double, or mix ]
tempDirectory, auto [ Possible value: DirName ]
useBlockingSync, auto [ Possible values: yes or no ]
useCpuPme, auto [ Possible values: yes or no ]
```

```
    OpenCL:

    deviceIndex, auto  [ Possible values: 0, '0 1' etc. ]
    openCLPlatformIndex, auto  [ Possible value: Number]
    precision, single  [ Possible values: single, double, or mix ]
    useCpuPme, auto [ Possible values: yes or no ]
```

A brief description of parameters is provided below:

```
    CPU:

    threads: Number of threads to use for simulation.

    CUDA:

    deviceIndex: Space delimited list of device indices to use for
        calculations.
    deterministicForces: Generate reproducible results at the cost of a
        small decrease in performance.
    precision: Number precision to use for calculations.
    tempDirectory: Directory name for storing temporary files.
    useBlockingSync: Control run-time synchronization between CPU and
        GPU.
    useCpuPme: Use CPU-based PME implementation.

    OpenCL:

    deviceIndex: Space delimited list of device indices to use for
        simulation.
    openCLPlatformIndex: Platform index to use for calculations.
    precision: Number precision to use for calculations.
    useCpuPme: Use CPU-based PME implementation.
```

-r, --restart <yes or no> [default: no]

Restart simulation using a previously saved final state checkpoint or XML file.

--restartParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs for restarting a simulation.

The supported parameter names along with their default values are are shown below:

```
    finalStateFile, <OutfilePrefix>_FinalState.<chk>  [ Possible values:
        Valid final state checkpoint or XML filename ]
    dataAppend, yes [ Possible values: yes or no]
```

A brief description of parameters is provided below:

```
    finalStateFile: Final state checkpoint or XML file
    dataAppend: Append data to existing trajectory and data log files
        during the restart of a simulation using a previously saved  final
        state checkpoint or XML file.
```

--restraintAtoms <yes or no> [default: no]

Restraint atoms during a simulation. The motion of specified atoms is restricted by adding a harmonic force that binds them to their starting positions. The atoms are not completely fixed unlike freezing of atoms. Their motion, however, is restricted and they are not able to move far away from their starting positions during local energy minimization and MD simulation.

--restraintAtomsParams <Name,Value,..>

A comma delimited list of parameter name and value pairs for restraining atoms during a simulation. You must specify these parameters for 'yes' value of '--restraintAtoms' option.

The supported parameter names along with their default values are are shown below:

```
    selection, none [ Possible values: CAlphaProtein, Ions, Ligand,
        Protein, Residues, or Water ]
    selectionSpec, auto [ Possible values: A space delimited list of
        residue names ]
    negate, no [ Possible values: yes or no ]
```

A brief description of parameters is provided below:

```
selection: Atom selection to restraint.
selectionSpec: A space delimited list of residue names for
    selecting atoms to restraint. You must specify its value during
    'Ligand' and 'Protein' value for 'selection'. The default values
    are automatically set for 'CAlphaProtein', 'Ions', 'Protein',
    and 'Water' values of 'selection' as shown below:

    CAlphaProtein: List of stadard protein residues from pdbfixer
        for selecting CAlpha atoms.
    Ions: Li Na K Rb Cs Cl Br F I
    Water: HOH
    Protein: List of standard protein residues from pdbfixer.

negate: Negate atom selection match to select atoms for freezing.
```

In addition, you may specify an explicit space delimited list of residue names using 'selectionSpec' for any 'selection". The specified residue names are appended to the appropriate default values during the selection of atoms for restraining.

--restraintSpringConstant <number> [default: 2.5]

Restraint spring constant for applying external restraint force to restraint atoms relative to their initial positions during 'yes' value of '--restraintAtoms' option. Default units: kcal/mol/A**2. The default value, 2.5, corresponds to 1046.0 kjoules/mol/nm**2. The default value is automatically converted into units of kjoules/mol/nm**2 before its usage.

--simulationParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs for simulation.

The supported parameter names along with their default values are are shown below:

```
steps, 1000000 [ Possible values: > 0. The default value
    corresponds to 4 ns for step size of 4 fs. ]

minimization, yes [ Possible values: yes or no ]
minimizationMaxSteps, auto  [ Possible values: >= 0. The value of
    zero implies until the minimization is converged. ]
minimizationTolerance, 0.24  [ Units: kcal/mol/A. The default value
    0.24, corresponds to OpenMM default of value of 10.04
    kjoules/mol/nm. It is automatically converted into OpenMM
    default units before its usage. ]

equilibration, yes [ Possible values: yes or no ]
equilibrationSteps, 1000  [ Possible values: > 0 ]
```

A brief description of parameters is provided below:

```
steps: Number of steps for production run.

minimization: Perform minimization before equilibration and
    production run.
minimizationMaxSteps: Maximum number of minimization steps. The
    value of zero implies until the minimization is converged.
minimizationTolerance: Energy convergence tolerance during
    minimization.

equilibration: Perform equilibration before the production run.
equilibrationSteps: Number of steps for equilibration.
```

-s, --smallMolFile <SmallMolFile>

Small molecule input file name. The macromolecue and small molecule are merged for simulation and the complex is written out to a PDB file.

--smallMolID <text> [default: LIG]

Three letter small molecule residue ID. The small molecule ID corresponds to the residue name of the small molecule and is written out to a PDB file containing the complex.

--systemParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs to configure a system for simulation.

The supported parameter names along with their default values are are shown below:

```
constraints, BondsInvolvingHydrogens [ Possible values: None,
    WaterOnly, BondsInvolvingHydrogens, AllBonds, or
    AnglesInvolvingHydrogens ]
constraintErrorTolerance, 0.000001
ewaldErrorTolerance, 0.0005

nonbondedMethodPeriodic, PME [ Possible values: NoCutoff,
    CutoffNonPeriodic, or PME ]
nonbondedMethodNonPeriodic, NoCutoff [ Possible values:
    NoCutoff or CutoffNonPeriodic]
nonbondedCutoff, 1.0 [ Units: nm ]

hydrogenMassRepartioning, yes [ Possible values: yes or no ]
hydrogenMass, 1.5 [ Units: amu]

removeCMMotion, yes [ Possible values: yes or no ]
rigidWater, auto [ Possible values: yes or no. Default: 'No' for
    'None' value of constraints; Otherwise, yes ]
```

A brief description of parameters is provided below:

```
constraints: Type of system constraints to use for simulation. These
    constraints are different from freezing and restraining of any
    atoms in the system.
constraintErrorTolerance: Distance tolerance for constraints as a
    fraction of the constrained distance.
ewaldErrorTolerance: Ewald error tolerance for a periodic system.

nonbondedMethodPeriodic: Nonbonded method to use during the
    calculation of long range interactions for a periodic system.
nonbondedMethodNonPeriodic: Nonbonded method to use during the
    calculation of long range interactions for a non-periodic system.
nonbondedCutoff: Cutoff distance to use for long range interactions
    in both perioidic non-periodic systems.

hydrogenMassRepartioning: Use hydrogen mass repartioning. It
    increases the mass of the hydrogen atoms attached to the heavy
    atoms and decreasing the mass of the bonded heavy atom to
    maintain constant system mass. This allows the use of larger
    integration step size (4 fs) during a simulation.
hydrogenMass: Hydrogen mass to use during repartioning.

removeCMMotion: Remove all center of mass motion at every time step.
rigidWater: Keep water rigid during a simulation. This is determined
    automatically based on the value of 'constraints' parameter.
```

--waterBox <yes or no> [default: no]

Add water box.

--waterBoxParams <Name,Value,..> [default: auto]

A comma delimited list of parameter name and value pairs for adding a water box.

The supported parameter names along with their default values are are shown below:

```
model, tip3p [ Possible values: tip3p, spce, tip4pew, tip5p or
    swm4ndp ]
mode, Padding  [ Possible values: Size or Padding ]
padding, 1.0
size, None  [ Possible value: xsize ysize zsize ]
shape, cube  [ Possible values: cube, dodecahedron, or octahedron ]
ionPositive, Na+ [ Possible values: Li+, Na+, K+, Rb+, or Cs+ ]
ionNegative, Cl- [ Possible values: Cl-, Br-, F-, or I- ]
ionicStrength, 0.0
```

A brief description of parameters is provided below:

```
model: Water model to use for adding water box. The van der
```

```
            Waals radii and atomic charges are determined using the
            specified water forcefield. You must specify an appropriate
            water forcefield. No validation is performed.
        mode: Specify the size of the waterbox explicitly or calculate it
            automatically for a macromolecule along with adding padding
            around ther macromolecule.
        padding: Padding around a macromolecule in nanometers for filling
            box with water. It must be specified during 'Padding' value of
            'mode' parameter.
        size: A space delimited triplet of values corresponding to water
            size in nanometers. It must be specified during 'Size' value of
            'mode' parameter.
        ionPositive: Type of positive ion to add during the addition of a
            water box.
        ionNegative: Type of negative ion to add during the addition of a
            water box.
        ionicStrength: Total concentration of both positive and negative
            ions to add excluding the ions added to neutralize the system
            during the addition of a water box.
```

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To perform a MD simulation for a macromolecule in a PDB file by using an NPT ensemble, applying system constraints for bonds involving hydrogens along with hydrogen mass repartioning, using a step size of 4 fs, performing minimization until it's converged along with equilibration for 1,000 steps (4 ps), performing production run for 1,000,000 steps (4 ns), writing trajectory file every 10,000 steps (40 ps), writing data log file every 1,000 steps (4 ps), generating a checkpoint file after the completion of the calculation, and generating a PDB for the final system, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  --waterBox yes
```

To run the first example for performing OpenMM simulation using multi- threading employing all available CPUs on your machine and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  --waterBox yes --platformParams "threads,0"
```

To run the first example for performing OpenMM simulation using CUDA platform on your machine and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  --waterBox yes -p CUDA
```

To run the second example for performing NPT simulation minimizing for a maximum of 2,000 steps, performing production run of 10,000 steps (40 ps), writing trajectory file every 1,000 steps (4 ps), and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  --waterBox yes --platformParams "threads,0"
  --simulationParams "steps,10000, minimizationMaxSteps, 1000"
  --outputParams "trajSteps,1000"
```

To run the second example for a marcomolecule in a complex with a small molecule and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  -s Sample13Ligand.sdf --waterBox yes --platformParams "threads,0"
```

To run the second example for performing NVT simulation and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  -s Sample13Ligand.sdf --mode NVT --platformParams "threads,0"
```

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  -s Sample13Ligand.sdf --mode NVT --waterBox yes
  --platformParams "threads,0"
```

To run the second example for a macromolecule in a lipid bilayer, write out reimaged and realigned trajectory file for the periodic system, along with a PDB file corresponding to the first frame, and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample12LipidBilayer.pdb
   -o Sample12LipidBilayerMDSimulation
  --platformParams "threads,0" --integratorParams
  "barostat,MonteCarloMembrane"
```

To run the second example by freezing CAlpha atoms in a macromolecule without using any system constraints to avoid any issues with the freezing of the same atoms, using a step size of 2 fs, and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  --waterBox yes --freezeAtoms yes
  --freezeAtomsParams "selection,CAlphaProtein"
  --systemParams "constraints, None"
  --platformParams "threads,0" --integratorParams "stepSize,2"
```

To run the second example by restrainting CAlpha atoms in a macromolecule and and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  --waterBox yes --restraintAtoms yes
  --restraintAtomsParams "selection,CAlphaProtein"
  --platformParams "threads,0" --integratorParams "stepSize,2"
```

To run the second example for a marcomolecule in a complex with a small molecule by using implicit water and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -i Sample13.pdb -o Sample13MDSimulation
  -s Sample13Ligand.sdf --mode NVT --platformParams "threads,0"
  --forcefieldParams "biopolymer, amber14-all.xml, water,
  implicit/obc2.xml"
```

To run the second example by specifying explict values for various parametres and generate various output files, type:

```
% OpenMMPerformMDSimulation.py -m NPT -i Sample13.pdb
  -o Sample13MDSimulation
  -f " biopolymer,amber14-all.xml,smallMolecule, openff-2.2.1,
  water,amber14/tip3pfb.xml" --waterBox yes
  --integratorParams "integrator,LangevinMiddle,randomSeed,42,
  stepSize,2, temperature, 300.0,pressure, 1.0"
  --outputParams "checkpoint,yes,dataLog,yes,dataStdout,yes,
  minimizationDataStdout,yes,minimizationDataLog,yes,
  pdbOutFormat,CIF,pdbOutKeepIDs,yes,saveFinalStateCheckpoint, yes,
  traj,yes,xmlSystemOut,yes,xmlIntegratorOut,yes"
  -p CPU --platformParams "threads,0"
  --simulationParams "steps,10000,minimization,yes,
  minimizationMaxSteps,500,equilibration,yes,equilibrationSteps,1000"
  --systemParams "constraints,BondsInvolvingHydrogens,
  nonbondedMethodPeriodic,PME,nonbondedMethodNonPeriodic,NoCutoff,
  hydrogenMassRepartioning, yes"
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

OpenMMExecuteMDSimulationProtocol.py, OpenMMPrepareMacromolecule.py, OpenMMPerformMinimization.py, OpenMMPerformSimulatedAnnealing.py

COPYRIGHT

Copyright (C) 2026 Manish Sud. All rights reserved.

The functionality available in this script is implemented using OpenMM, an open source molecuar simulation package.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.