

# EFPMD manual

July 8, 2024

## Introduction to EFPMD

EFPMD is a molecular simulation program based on LIBEFP. It supports single point energy and gradient calculations, semi-numerical Hessian and normal mode analysis, geometry optimization, molecular dynamics simulations in microcanonical (NVE), canonical (NVT), and isobaric-isothermal (NPT) ensembles.

Simulations can be accelerated by running in parallel mode on multi-core CPUs. To enable parallel computation set `OMP_NUM_THREADS` environment variable to the desired number of parallel threads. For example, on a 4-core machine you can set the `OMP_NUM_THREADS` environment variable to four before starting the program:

```
export OMP_NUM_THREADS=4
```

This will enable parallel computation using 4 threads and should give almost 4x speedup in all calculations. If you have compiled *libefp* with MPI support you can also run parallel calculations across multiple nodes using standard *mpirun* command:

```
mpirun -np 8 efpmd input.in
```

Note that you can achieve better scalability by using OpenMP for parallelization within a single node and MPI for inter-node communication. Additional examples of input files can be found in the `tests` directory in the source code archive.

## Input file format

It comprises of two parts: (1) the parameter section containing the **runtype**, **coordinate type**, **energy terms** and other parameters specifying the property desired and (2) the geometry section.

```
run_type sp
coord xyzabc
terms elec pol disp xr
disp_damp overlap
fraglib_path ../fraglib

fragment h2o_1
0.000  0.000  0.000  0.000  0.000  0.000
fragment h2o_1
5.000  0.000  0.000  2.000  0.000  2.000
fragment nh3_1
0.000  5.000  0.000  5.000  0.000  -3.000
```

## Generic parameters

### Type of simulation

`run_type [xyzabc|points|rotmat|atoms]`

- **sp** - single point energy calculation.
- **grad** - energy gradient calculation.
- **hess** - semi-numerical Hessian calculation and normal mode analysis.
- **opt** - geometry optimization.
- **md** - molecular dynamics simulation.
- **efield** - compute and print electric field on all atoms.
- **elpot** - compute and print electrostatic potential on all atoms.
- **gtest** - compute and compare numerical and analytical gradients.
- **etest** - compute and compare total energy.

Default value: `sp`

## Format of Fragment Input

`coord [xyzabc|points|rotmat|atoms]`

- **xyzabc** - Coordinates of the center of mass and Euler angles.
- **points** - Coordinates of three atoms for each fragment.
- **rotmat** - Coordinates of the center of mass and rotation matrix.
- **atoms** - Atom names and cartesian coordinates of all atoms.

Default value: `points`

See fragment input specification for more details. **ATTENTION:** Default changed to `points` from `xyzabc` in version 1.7.3.

## Energy Terms for EFP Computation

`terms [elec|pol|disp|xr]`

- **elec** - Include electrostatics energy.
- **pol** - Include polarization energy.
- **disp** - Include dispersion energy.
- **xr** - Include exchange repulsion energy.

Default value: `elec pol disp xr`

## Electrostatic Damping Type

`elec_damp [screen|overlap|off]`

- **screen** - Damping formula based on SCREEN group in the EFP potential.
- **overlap** - Overlap-based damping formula. This damping correction is printed as charge penetration energy.
- **off** - No electrostatic damping.

Default value: `screen`

### Dispersion Damping Type

`disp_damp [tt|overlap|off]`

- **tt** - Damping based on the formula by Tang and Toennies.
- **overlap** - Overlap-based dispersion damping.
- **off** - No dispersion damping.

Default value: `overlap`

### Polarization Damping Type

`pol_damp [tt|off]`

- **tt** - Tang and Toennies like damping formula.
- **off** - No polarization damping.

Default value: `tt`

### Polarization Solver

`pol_driver [iterative|direct]`

- **iterative** - Iterative solution of system of linear equations for polarization induced dipoles.
- **direct** - Direct solution of system of linear equations for polarization induced dipoles. This solver does not have convergence issues but is unsuitable for large systems (more than 2000 polarizable points). The direct solver is not parallelized.

Default value: `iterative`

### Enable Molecular-Mechanics Force-Field for Flexible EFP Links

`enable_ff [true|false]`

Default value: `false`

### Enable Multistep Molecular Dynamics

`enable_multistep [true|false]`

Default value: `false`

## Geometry of the Molecular-Mechanics Part

`ff_geometry <path>`

Default value: **ff.xyz**

## Molecular-Mechanics Force-Field Parameters File

`ff_parameters <path>`

Default value: **fraglib/params/amber99.prm**

## Use Single EFP Parameters File

`single_params_file [true|false]`

Default value: **false**

## Single EFP Parameters File Path

`efp_params_file <path>`

Default value: **params.efp**

## Enable cutoff for fragment/fragment interactions

`enable_cutoff [true|false]`

Default value: **false**

## Cutoff distance for fragment/fragment interactions

`swf_cutoff <value>`

Default value: **10.0**

Unit: Angstrom

## Cutoff distance for exchange-repulsion interactions between fragments

`xr_cutoff <value>`

Default value: **swf\_cutoff**

Unit: Angstrom

## Maximum number of steps to make

`max_steps <number>`

Default value: **100**

This specifies maximum number of steps for both geometry optimization and molecular dynamics.

## Number of steps between recomputation of slow terms in multistep MD

`multistep_steps <number>`

Default value: 1

Currently only exchange-repulsion EFP term is affected.

## The path to the directory with fragment library

`fraglib_path <path>`

Default value: `"$(prefix)/share/libefp"` (data install directory)

The `<path>` parameter should not contain spaces or should be inside double quotes otherwise.

## The path to the directory with user-created fragments

`userlib_path <path>`

Default value: `"."` (current directory)

The `<path>` parameter should not contain spaces or should be in double quotes otherwise.

# Pairwise energy analysis

## Enable/Disable pairwise analysis

`enable_pairwise [true|false]`

Default value: **false**

## Specify ligand

`ligand [integer]`

Default value: 0 (the first fragment in the system)

# Periodic Boundary Conditions (PBC)

## Enable/Disable PBC

`enable_pbc [true|false]`

Default value: **false** Setting `enable_pbc` to **true** also sets `enable_cutoff` to **true**.

### Periodic Box Size

```
periodic_box <x> <y> <z> <alpha> <beta> <gamma>
```

Default value: 30.0 30.0 30.0 90.0 90.0 90.0

Unit: Angstroms, degrees

If only three values are given, the angles are set to 90 degrees (orthogonal box). Non-orthogonal PBC is implemented only for single-point energy calculations. The smallest box dimension must be greater than  $2 \times \text{swf\_cutoff}$ .

### Print PBC coordinates

```
print_pbc [true|false]
```

Default value: **false**

Prints coordinates of the system contained in a single periodic cell around a fragment specified by **ligand** keyword.

### Use symmetry

```
symmetry [true/false]
```

Default value: **false**

If true, effectively performs calculations only on symmetry-unique fragments, which speeds up calculations of symmetric crystal systems with PBC. Implemented for single-point energy calculations. Not parallelized. See **symm\_frag** keyword for specifying symmetry-identical fragments.

### Specifying symmetry-identical fragments

```
symm_frag [frag | list]
```

Default value: **frag**

**frag** - assumes that all fragments of the same type are identical.

**list** - not implemented.

## Geometry optimization related parameters

### Optimization tolerance

```
opt_tol <value>
```

Default value: **0.0001**

Unit: Hartree/Bohr

Optimization will stop when the maximum gradient component is less than `opt_tol` and RMS gradient is less than one third of `opt_tol`.

## Gradient test related parameters

See also `num_step_dist` and `num_step_angle`.

### Test tolerance

`gtest_tol <value>`

Default value: **1.0e-6**

Unit: Hartree/Bohr

### Reference energy value

`ref_energy <value>`

Default value: **0.0**

Unit: Hartree

## Hessian calculation related parameters

### Hessian accuracy

`hess_central [true|false]`

Default value: **false**

If `hess_central` is **true**, then the more accurate central differences will be used for numerical Hessian calculation. Otherwise, only forward differences will be used. Note that central differences require twice as many gradient calculations.

### Numerical differentiation step length for distances

`num_step_dist <value>`

Default value: **0.001**

Unit: Angstrom



### Numerical differentiation step length for angles

`num_step_angle <value>`

Default value: **0.01**

Unit: Radian

## Molecular dynamics related parameters

### Ensemble

`ensemble [nve|nvt|npt]`

- **nve** - Microcanonical ensemble.
- **nvt** - Canonical ensemble with Nose-Hoover thermostat. For the description of the algorithm see [Phys. Rev. A 31, 1695 \(1985\)](#).
- **npt** - Isobaric-isothermal ensemble. This also sets `enable_pbc` to true. For the description of the algorithm see [Mol. Phys. 78, 533 \(1993\)](#).

Default value: **nve**

### Time step

`time_step <value>`

Default value: **1.0**

Unit: Femtosecond

### Print step

`print_step <value>`

Default value: **1**

Number of steps between outputs of the system state.

### Assign initial velocities

`velocitize [true|false]`

Default value: **false**

If true, then random initial velocities will be assigned to fragments using Gaussian distribution. Velocity magnitudes are chosen so that the initial temperature of the system is approximately equal to the target simulation temperature.

### Simulation temperature

`temperature <value>`

Default value: **300.0**

Unit: Kelvin

Target simulation temperature for NVT and NPT thermostats.

### Simulation pressure

`pressure <value>`

Default value: **1.0**

Unit: Bar

Target simulation pressure for NPT barostat. Note that whether or not pressure coupling is used, the pressure value will oscillate significantly. Fluctuations on the order of hundreds of bar are typical. This variation is entirely normal due to the fact that pressure is a macroscopic property and can only be measured properly as a time average. The actual variations of instantaneous pressure depend on the size of the system and the values of barostat parameters.

### Thermostat parameter

`thermostat_tau <value>`

Default value: **1000.0**

Unit: Femtosecond

Temperature relaxation time parameter of the Nose-Hoover thermostat.

### Barostat parameter

`barostat_tau <value>`

Default value: **10000.0**

Unit: Femtosecond

Pressure relaxation time parameter of the barostat.

## Fragment input

One or more fragment `<name>` groups.

If `<name>` contains an `_1` suffix, the parameters (`.efp` files) for this fragment will be searched for in the `fraglib_path` directory. Otherwise, the

directory specified by the `userlib_path` option will be used. The name of the `.efp` file must be the same as the name of the fragment without an `_l` suffix and must be all lowercase. For example, for the fragment named `H2O_L`, the parameters must be in the `fraglib_path` directory in the file named `h2o.efp`. For the fragment named `NH3`, the parameters must be in the `userlib_path` directory in the file named `nh3.efp`. Fragment position and orientation are specified on the next line(s).

#### **Format of input when `coord` is `xyzabc`**

---

```
fragment h2o
0.000  0.000  0.000  0.000  0.000  0.000
```

The numbers are coordinates of the center of mass of a fragment in Angstroms and three Euler rotation angles in Radians.

#### **Format of input when `coord` is `points`**

---

```
fragment h2o
0.0 0.0 0.0
1.0 0.0 0.0
0.0 1.0 0.0
```

The numbers are coordinates of three points belonging to a fragment in Angstroms.

#### **Format of input when `coord` is `rotmat`**

---

```
fragment h2o
0.0 0.0 0.0
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
```

The numbers are coordinates of the center of mass of a fragment in Angstroms and a 3 x 3 rotation matrix.

#### **Fragment velocities**

Fragment velocities for MD simulations can be specified using `velocity` keyword with the center of mass velocity and angular velocity in atomic units specified on the next line.

### **Fragment constraints**

Quadratic constraint on the fragment center of mass can be specified using `constraint` keyword with the force constant  $k$  (in a.u.) and constraint position  $xyz$  (in angstroms) specified on the next line.

## **Input of point charges**

Additionally to fragments a system can contain a set of point charges. They can be specified using the following format for each charge:

```
charge <q> <x> <y> <z>
```

### **Print additional information for debugging**

```
print <value>
```

Default value: **0**

0 - standard output, 1 - more information, 2 - even more information.