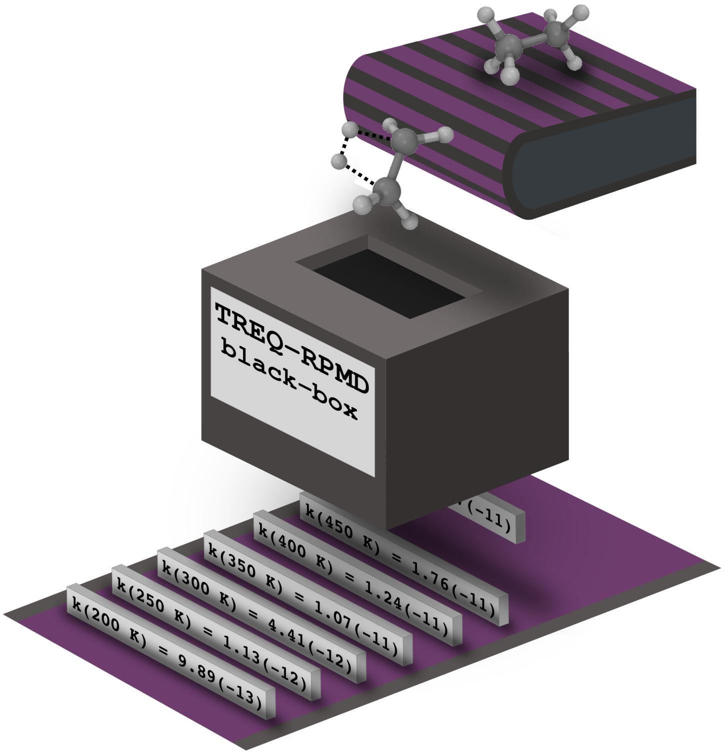
EVB-QMDFF

RPMD and Rate Constant Calculations on Black-Box Potential Energy Surfaces



Julien Steffen

Version 1.1, September 2021

Contents

1 General Information

EVB-QMDFF is a program package for molecular dynamics simulations. It can both perform classical MD and ring polymer molecular dynamics (RPMD) on potential energy surfaces built upon the quantum mechanically derived force field (QMDFF) by Grimme and the empirical valence bond method by Warshel and others.

QMDFF is able to generate a fully-fledged potential energy description for an arbitrary chemical system, leading to a reasonable description of the accessible energy landscape as well as of bond breakings. With this ability, one can e.g. simulate mechanochemical reaction with a single QMDFF. Two or three QMDFF can be coupled via EVB in order to simulate arbitrary chemical elementary reactions. Based on this simple idea, a list of EVB coupling methods, including the newly-developed transition path corrected reaction path EVB-QMDFF (TREQ) was implemented into the program package. Based upon umbrella samplings of the reaction path, a protocol for black-box calculations of chemical reaction rate constants was implemented, serving as a simple and powerful tool for the calculation of those values.

A second branch is the simulation of liquid phases via QMDFF. It is possible to generate a force field for a liquid of arbitrary composition from QMDFFs of the single components. The targeted optimization of noncovalent QMDFF parameters is now possible as well, leading to force fields of high quality for the liquid of interest, which can be sampled via NVT or NpT dynamics classically or with RPMD.

EVB-QMDFF consists of a number of independent programs, serving to fulfill a number of different tasks:

- **qmdffgen**: Generates a QMDFF from given output of a QM calculation containing the bond orders, charges and frequencies of the reference structure.

- **evbopt**: Optimizes the parameters of a chosen EVB coupling term (dE, dQ or DG coupling) for a chemical reaction based on two QMDFFs (reactand and product) and reference energies of the reaction path.

- **egrad**: Calculates energies and gradients for a given trajectory of structures with one of the given PES descriptions.

- **evb_qmdff**: Performs geometry optimizations, gradient and frequency calculations of chemical systems with one of the given PES descriptions.

- **irc**: Optimizes the transition state and, starting from it, a reaction path with the IRC method.
- **dynamic**: Molecular dynamics simulations, both classical and RPMD, on QMDFF and EVB-QMDFF potential energy surfaces. Forces can be applied on different collective variables, for sampling of certain chemical processes.
- **rpmd**: Calculation of chemical reaction rate constants based on RPMD on EVB-QMDFF potential energy surfaces.
- **evb_kt_driver**: With a reaction path as input, reference calculations, PES setup with TREQ and RPMD samplings are done in black-box fashion, resulting in temperature-dependent reaction rate constants and Arrhenius parameters.
- **mult_qmdff**: The QMDFF of a solvent box is built from given QMDFFs of single solvent molecules.
- **qmdffopt**: After automatic builtup of a suited trainset, noncovalent QMDFF parameters can be optimized for better solvent properties.

2 Installation

EVB-QMDFF consists of pure Fortran code and thus is quite easy to install.

2.1 Prerequisites

All you need is:

- A Fortran compiler (`gfortran` or `ifort`)
- `BLAS` and `LAPACK` linear algebra libraries
- `FFTW3` Fourier transform libraries
- Some of the programs (**evbopt** and **rpmd**) can be run in parallel using MPI. If you want to compile and run them in parallel, `MPI` must be installed.

2.2 Settings

After downloading EVB-QMDFF, go into the `EVB-QMDFF` main folder and open the `Makefile` located there. In its header, the Fortran compilers (GNU or Intel) can be chosen:

```
#  Fortran compiler for serial version:
FC = gfortran  # GNU Fortran
# FC = ifort   # Intel Fortran compiler
#  Fortran compiler for MPI version
# FC = mpif90  # GNU Fortran
# FC = mpifort # Intel Fortran compiler
```

The serial GNU compiler is chosen as default, please uncomment your preferred version.
By default, the single program binaries are located both into the `bin` subdirectory of the program folder and the bin directory in your home (`/bin`). Plase edit this part right below the compiler settings to your preferred location:

```
 #  Location of the compiled and linked executables
BINDIR = ~/bin
```

Now save the Makefile and copy it to the `src` subdirectory.

2.3 Compilation

In the `src` subdirectory, execute:

```
make
```

and the objects and executables will be compiled. Now, EVB-QMDFF is ready to use.

3 Program handling

In this chapter, the handling of each program will be explained. For each of them, at least one example is located into the `examples` folder of the EVB-QMDFF directory. Switch to the subfolders named after the respective programs in order to calculate the examples prepared for you if you want to get more experienced in using EVB-QMDFF.
Besides the descriptions provided below, each program has a help function, that gives general information and a complete list of availabe keywords for handling. It can be invoked with:

```
[program.x] -h
```

or

```
[program.x] -help
```

For almost all programs, the main input file is a `[name].key` file, where all needed commands to execute your calculation are listed.

3.1 qmdffgen

3.1.1 Introduction

qmdffgen uses the output of an QM calculation for a molecule to generate a QMDFF for it. This QMDFF is able to simulate the full configurational landscape of this molecule and further to describe bond breakings correctly. A QMDFF needs a predefine set of information provided by the QM program:

- Optimized geometry
- Wiberg-Mayer bond orders
- Hirshfeld charges
- Hessian matrix

Provided the correct settings, all those can be obtained from a single QM calculation (see below). After doing the QM calculation, qmdffgen can be invoked:

```
qmdffgen.x
```

The program is interative and asks you for the Software used for the reference calculation (currently available: orca, Gaussian, Turbomole and CP2K), the number of QMDFFs to generate (one to three), and the prefix of the QM input files (name without file ending). Then, the QMDFF is generated in a black-box procedure and directly useable for other calculations!

3.1.2 Example 1: Octane with Orca

Folder: `qmdff/ocane_orca/`

The input as well as the output of the QM reference calculation with orca is placed here. Basically, the keywords for the reference calculation with orca are always the same. Open the file `octane.inp`:

```
 ! PBE0 D3 def2-SVP opt freq
* xyz 0 1
C         -8.36972        0.54078       -0.36595
C         -6.84957        0.56234       -0.35222
...
H         -0.30022        5.11409        2.54651
H         -0.26113        4.29744        0.97442
*
%output
        Print[ P_Hirshfeld ] 1
end
```

A geometry optimization of the octane molecule was done with the `PBE0` DFT functional (with `Grimme` D3 dispersion correction) and the cheap `def2-SVP` basis set. For useful output, a geometry optimization `opt` followed by a frequency calculation `freq` need to be done, further, the section

```
 %output
        Print[ P_Hirshfeld ] 1
end
```

must be added for the calculation of Hirshfeld charges. The **qmdffgen** program needs the `octane.hess` and `octane.out` files as reference.
Since these are present, simply type

```
qmdffgen.x
```

and give the respective commands during the interactive input. Then, the QMDFF `octane.qmdff` will be generated.
During the calculation, the central quality parameters are printed into the terminal:

```
Zero point vibr. energy comparison (FF/true, kcal)  :     150.432     154.650
Mean absolute deviation (MAD) of frequencies (cm-1) :      54.560
```

The MAD value in particular is a good indicator of the QMDFF-quality. It is calculated by performing frequency calculations for the QM reference and the QMDFF hessian and comparing them. Good QMDFFs achieve values between 20 and 50, if the value is above 100, something might went wrong during the QM reference calculation or the optimization. If the error cannot be located, please let me know. The QMDFF was written into `octane.qmdff`. This file is the entrypoint for all the other programs included into EVB-QMDFF.
A number of additional output files was produced during the QMDFF optimization. these are:

- `octane_qmdff.log`: Detailed information about the QMDFF generation (QMDFF internal coordinates, fragments, Levenberg-Marquardt fit of the force constants and QMDFF quality). See the QMDFF paper [**?**] for more details about the whole procedure.

- `octane_molden.out`: This file can be opened with the `molden` molecule viewer in order to visualize the normal modes of the different frequencies of the QMDFF hessian.

- `coord_def.inp`: List of internal coordinates used to construct the QMDFF (bonds, angles, dihedrals) in the formate also used for several other programs inside QMDFF (**irc**, **rpmd**, ...).

- `coord_analysis.dat`: Values of all internal coordinates, of the Wilson matrix for transformation of cartesian gradients into intenal coordinates and the Wilson matrix derivative for the transformation of cartesian Hessians into internal coordinates. These matrices are used for setup of the DG-EVB coupling term in **evbopt**.

3.1.3 Example 2: Guaicol with Gaussian

Folder: `qmdff/guaicol_gaussian/`

For Gaussian, geometry optimization and frequency calculation must be done separately, since a combined geopt+freq calculation writes out the Hessian in an incomplete formate.Therefore, the file `guaicol_opt.gjf` performs the geometry optimization and the file `guaicol_freq.gjf` performs the calculation of the Hessian.
The QMDFF can again be calculated by generated by typing:

```
qmdffgen.x
```

Note, that the prefix of the QM reference must be that of the frequency calculation, i.e., `guaicol_freq`.
The quality of the generated QMDFF can be seen from the lines:

```
 Zero point vibr. energy comparison (FF/true, kcal)  :      85.684      86.016
Mean absolute deviation (MAD) of frequencies (cm-1) :      26.815
```

The MAD value of 26 suggests a good description of the low-energy part of the potential energy surface of Guaicol with the generated QMDFF (which was written to `guaicol_freq.qmdff`. The other output files are the same as explained in the first example.

3.2 evbopt

3.2.1 Introduction

This program generates and optimizes the parameters of EVB coupling terms. The coupling smoothly connects two QMDFFs, therefore these need to be present (generated with **qmdffgen**) when an EVB coupling term is optimized. In addition, some reference data describing the reaction must be present, serving as the training set for the optimization routine. The xyz-coordinates of the structures and the respective reference energies in a linewise format must be present. Usually, a reaction path, generated from a NEB or IRC calculation, would be used for this. For the RP-EVB and the related TREQ method, a reaction path is mandatory.

3.2.2 dE-EVB

Folder: `evbopt/dE-EVB/`

The simplest EVB coupling version is the energy-gap (dE)-EVB coupling. Here, the off-diagonal coupling term depends only on the energetic distances of both QMDFFs at the

current coordinates ($C(\Delta E)$. A bunch of different functions depending on ΔE can be used, the easiest are the constant coupling $C = a$ with no ΔE-dependence at all and the `1g` coupling $C(\Delta E) = a \exp{(-b(\Delta E)^2)}$ with a Gaussian shape.
The main input file is the `evbopt_de_1g.key` file, where all essentially all control parameters are listed:

```
PES de_evb
QMDFFNAMES  min1.qmdff  min2.qmdff
ESHIFT    -272.22659664700754     -272.16798367875066
energies_ref ref.dat
coords_ref struc.xyz
shift_manual
de_evb{
   coupling sp
}
```

The first keyword is always needed and specifies the used class of PES functions. Here, we want to use the dE-EVB coupling (`de_evb`). Other options (for **evbopt**) are dQ-EVB (`dq_evb`) and DG-EVB (`dg_evb`), see below.
The keywords `QMDFFNAMES` and `ESHIFT` specifiy the properties of the single QMDFFs, both their filenames and the energy shifts. These parameters are written automatically if **qmdffgen** is executed first (see section **??**).
Both the `ENERGIES_REF` and `COORDS_REF` keywords are needed to provide the needed reference data for the reaction of interest that shall be described with EVB. Usually this would be a one-dimensional reaction path, but higher dimensional sampling data could be used as well.
The optimization is invoked by typing:

```
evbopt.x evbopt.key
```

A multi-start local-search Levenberg-Marquardt optimization is executed with the objective to minimize the error sum between the EVB-QMDFF energies and the reference energies listed in `ref.dat`, an usual output is:

```
...
Used off-diagonal-Term:
```

```
a*exp(-b*deltaE^2)+c*deltaE*exp(-d*deltaE^2)   (sp)
After Multi-Start-Local-Search step:           1
Local optimization finished, new best fitness:   8.3897511547073188E-004
After Multi-Start-Local-Search step:          15
Local optimization finished, new best fitness:   8.3877256721804922E-004
After Multi-Start-Local-Search step:          35
Local optimization finished, new best fitness:   8.3632257233213895E-004
.. .----. -- -.. --- -. .
Multi start local optimization finished!
The best fitness-value is:   8.3632257233213895E-004
Look into evbopt.log for further details.
...
```

The total fitness is the central indicator of how good the reference reaction path is reproduced with the EVB-QMDFF model, however, in order to gain more detailed knowledge of the quality, one can look at the additional output, namely:

- `energies.qmdff`: The EVB-QMDFF energies of all reference structures
- `single_qmdff.dat`: The energies of both QMDFFs of all reference structure

Plotting the files together with the reference energies in `ref.dat` results in the figure shown in fig. 1

3.2.3 DG-EVB

Folder: `evbopt/DG-EVB/`

The distributed Gaussian (DG-EVB) coupling term enables the inclusion of QM-reference gradients and frequencies into the formulation of the transition region, offering a much better quality than dE-EVB or dQ-EVB.
The following explanation and the example assumes that an optimized 1D reaction path of the reaction of interest is given. In principle, however, arbitrary collections of reference data, e.g. from MD samplings, can be used as well.
Including gradients and frequencies to the used reference set needs somewhat more preparational work than using only energies of all structures of the reaction path. The first question is:

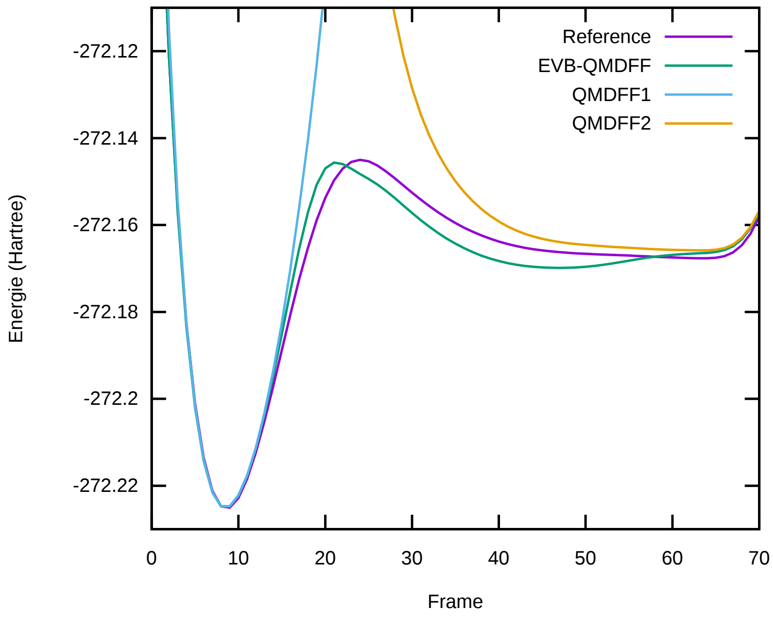


Figure 1: Reference energies plotted together with QMDFF energies and energies of the optimized dE-EVB-QMDFF.

how many and which points along the reaction path shall be chosen. In principle, we could simply choose all given structures and calculate gradients and frequencies on it. Depending on the resolution of the path, dozens to hundreds of frequency calculations might be needed, which almost always implies heavy computational demands. Further, the separate optimization of coupling basis function widths needs an additional set of energies along the path. In this example, the resulting input files are provided, such that this prepatational part might be skipped. It serves, however, as a template for how to prepare an DG-EVB-QMDFF surface for other reactions of interest,

For the manual preparation, an utility script for the selection and preparation of gradient/frequency reference is given in the `scripts` main folder. Copy the script `dg_evb_ref.sh` to the actual folder if you want to redo the reference calculations. It requires the reaction path stored in *struc.xyz* and a list of structure numbers on which the reference data shall be calculated: `struc_num.txt`. A file containing the header of the desired QM input (for Gaussan: `gauss.com`, for orca: `orca.inp`, for CP2K: `run.inp`) must be provided as well.

For convenience, the folder structure was already prepared for seven reference points distributed along the path such that the transition state and the important transition region is covered well. Gaussian is here the chosen QM program, the frequency calculations can be executed for all

seven folders.
After this, the needed reference input file dg_ref.dat can be generated by executing the program gen_dg_evb.
The main input file for EVB optimizations is evbopt.key. It has the following entries:

```
 PES dg_evb
QMDFFNAMES  min1.qmdff  min2.qmdff
ESHIFT  -132.07243067568406       -132.09078533339240
energies_ref ref.dat
coords_ref struc.xyz
shift_manual

dg_evb{
   mode 2
   points 7
   point_ref dg_ref.dat
   read_coord
}
```

After specification of a DG-EVB potential energy surfaces, the information concerning the single QMDFFs and the used reference data for optimization is listed. All settings concerning the DG-EVB coupling term in particular are listed inside the dg_evb block. They have the following meaning:

- mode: Which type of DG-EVB function built upon which kind of ererence shall be used. 1: only energies, 2: energies and gradients, 3 energies, gradients and Hessians.
- points: Number of DG-EVB reference points on which the energies, gradients and Hessians were calculated.
- point_ref: File in which the reference information for the DG-EVB points is stored.
- read_coord: If given, the defined internal coordinates are read in from the file coord_def.inp, else, they are determined automatically.

Altogether, 4 or 5 different input files are needed to perform the optimization of a DG-EVB coupling term (named as above):

- `evbopt.key`: Main input file
- `ref.dat`: File with energies of the reaction path for Gaussian exponent optimization.
- `struc.xyz`: File with structures of the reaction path.
- `dg_ref.dat`: Reference information for the DG-EVB points.
- `coord_def.inp`: Definition of internal coordinates for DG-EVB coupling (optional).

The `dg_ref.dat` file needs to have a predefined formate, which is generated by the `gen_dg_ref` program, given in the scripts folder. Essentially, energies, geometries, gradients and Hessians of all reference points are there listed in order, here for seven reference points along the path:

```
NPOINTS 7

NATOM  6

*POINT 1

ENERGY    -132.073392

GEOMETRY
  -0.2598966E+01  -0.1689562E+00  -0.2337807E-02  -0.2485496E+01   0.1654694E+01
   0.3883103E-01   0.2523122E+01  -0.1090708E+00  -0.8583873E-02   0.1909804E+01
   0.1018732E+00   0.1786691E+01   0.1266274E+01  -0.1242103E+01  -0.8934828E+00
   0.2439287E+01   0.1600681E+01  -0.8532495E+00
END

GRADIENT
   0.3391129E-02   0.5053883E-05  -0.3630294E-03   0.1683734E-03  -0.3711497E-05
   0.1950042E-03  -0.2993883E-02  -0.4142712E-05  -0.4566316E-04   0.3697710E-04
   0.4438535E-04   0.7573431E-04  -0.3227359E-03  -0.1882671E-04   0.1766393E-03
  -0.2798607E-03  -0.2275832E-04  -0.3868530E-04
END

HESSIAN
```

```
   0.1371910E-01   0.3332208E-01   0.2103093E-03  -0.4587440E-02  -0.3262215E-01
  -0.6211021E-03  -0.5506326E-02  -0.1298989E-02  -0.9317258E-05  -0.4769846E-03
....
END

*POINT 2
...
```

Depending on the chosen DG-EVB coupling mode, energies (and geometries) might be sufficient as well.

If internal coordinates for the coupling are given in the `coord_def.inp` file, a predefined formate must be obeyed as well: One coordinate per line, each defined by the involved atom indices: Two for a bond, three for an angle, and four for dihedral/out of plane. For the latter one, a fifth number indicating if dihedral or out of plane must be added *after* the atom indices, such that each dihedral/out of plane line consists of five numbers.

In the example system, the internal coordinate set defined in `coord_def.inp` consists of four bonds:

```
 1 5
 3 5
 1 3
 1 2
```

The choice of internal coupling coordinates is not exactly black-box. The usual rule is that coordinates involved in the reaction of interest are the most important, but the overall quality might significantly raise or fall if another assumingly unimportant coordinate is added or removed.

In the **evb_kt_driver** program explained below a more systematic and quite well working approach of finding the internal coordinates for the there used TREQ PES is implemented.

The actual optimization of the DG-EVB coupling term is very simple: Just type

```
 evbopt.x evbopt.key
```

and the MSLS-LM algorithm is executed to optimize the parameter. This will take some seconds to minutes.

The program output should look something like this:

```
...
Used DG-EVB mode: 3 - energies, gradients and hessians (E+G+H)
The dimension of the linear equation F=DB is          105
##################  QMDFFs!
 . ...- -... --.- -- -.. ..-. ..-.
dg_evb-parameters are optimized with Multi Start Local Search.
We will start         100 single optimization runs.
The coefficient matrix D for DG-EVB will be calculated analytically!
Multi Start Local Search algorithm:
---------------------------------------------------
After Multi-Start-Local-Search step:           1
Local optimization not converged, new best fitness:   1.5411906000070418E-006
After Multi-Start-Local-Search step:           2
Local optimization finished, new best fitness:   1.5075939192383248E-006
After Multi-Start-Local-Search step:          10
Local optimization finished, new best fitness:   1.3147886030294805E-006
After Multi-Start-Local-Search step:          11
Local optimization finished, new best fitness:   1.2190281156368069E-006
After Multi-Start-Local-Search step:          16
Local optimization finished, new best fitness:   1.1965456308064260E-006
After Multi-Start-Local-Search step:          30
Local optimization finished, new best fitness:   1.1388462232211951E-006
After Multi-Start-Local-Search step:          78
Local optimization finished, new best fitness:   1.1059136690605482E-006
.. .----. -- -.. --- -. .
Multi start local optimization finished!
The best fitness-value is:   1.1059136690605482E-006
Look into evbopt.log for further details.
...
```

The total fitness is the central indicator of how good the reference reaction path is reproduced with the EVB-QMDFF model, however, in order to gain more detailed knowledge of the quality, one can look at the additional output, namely:

- `energies.qmdff`: The EVB-QMDFF energies of all reference structures

- single_qmdff.dat: The energies of both QMDFFs of all reference structure

Plotting the files together with the reference energies in ref.dat results in the figure shown in fig. 1.