QM/MM Force Matching in CPMD

Manuel Dömer, Patrick Maurer, Ivano Tavernelli, and Ursula Röthlisberger[*]

Laboratoire de Chimie et Biochimie Computationelle, Ecole Polytechnique Fédérale de Lausanne, Lausanne, CH-1025, Switzerland

E-mail: ursula.roethlisberger@epfl.ch

This document has been prepared based on the PhD thesis

M. Dömer, *Approaches to Increase the Accuracy in Molecular Dynamics Simulations: From Empirical to Tight Binding and First Principles Methods*, Ecole Polytechnique Fédérale de Lausanne, **2013**.

to serve as a user guide for the QM/MM force matching implementation in the CPMD software package, starting with release 3.15.[1] Please cite the references indicated below with results obtained using the QM/MM force matching code.

Further information can be obtained from our website `http://lcbc.epfl.ch`, section Research - QM/MM Force Matching.

Contents

---

[*]To whom correspondence should be addressed

1 How to Cite

Published results obtained with the QM/MM force matching implementation in CPMD should cite the following articles:

- Maurer, P.; Hugosson, H. W.; Colombo, M. C.; Rothlisberger, U. J. Chem. Theory Comput. **2007**, 3, 628-639.

- Doemer, M.; Maurer, P.; Campomanes, P.; Tavernelli, I.; Rothlisberger, U. *submitted to J. Chem. Theory Comput.*

2 Methods

2.1 QM/MM Force Matching

The concept of the force-matching approach, as proposed in the seminal paper by Ercolessi *et al.*, [2] involves the minimisation of the force-dependend objective function defined as:

$$\lambda(\tau) = \frac{1}{3 \cdot L \cdot N} \sum_{l=1}^{L} \sum_{\alpha=1}^{N_l} |\mathbf{F}_{l\alpha}(\tau) - \mathbf{F}_{l\alpha}^0|^2 \tag{1}$$

where τ is the set of parameters in the interatomic classical or semi-empirical potentials that are to be optimised by reproducing the reference forces calculated by *ab inito* or DFT methods. L is the number of atomic configurations, N is the number of atoms, $\mathbf{F}_{l\alpha}(\tau)$ is the force on atom α in configuration l with the parameterisation τ and $\mathbf{F}_{l\alpha}^0$ is the reference force. The objective function can also be complemented by contributions from additional constraints, e.g. experimental data.

In our force matching scheme a FF is parameterised for all the QM atoms of a complete QM/MM system. In any QM/MM scheme, this relies on the assumption that the original FF parameters provide a good representation for the rest of the system, i.e. the MM region. The procedure involves four steps: First, a set of reference configurations l is generated. Second, the nuclear

forces on all the atoms of the QM subsystem as well as the electrostatic potential and field in the surrounding of the electronic charge density are stored. Third, a set of atomic point charges $\{q_\alpha\}$ that reproduce the electrostatic potential and field in the surrounding of the QM region is calculated. The fourth step involves the actual force matching. The non-bonded contributions, computed with the charges obtained in the third step and a given set of Lennard-Jones parameters from the MM force field, are subtracted from the total reference forces on the QM atoms. The remaining forces are assumed to originate from bonded interactions. The parameters for bonded interactions (torsions, bending, and bonds) are thus optimised in order to reproduce the residual part of the reference forces. The entire procedure and the implementation in the CPMD package is described in detail in the following sections.

2.1.1 Reference Configurations

We extract the reference configurations from a finite temperature QM/MM CPMD run at the same level of theory than the one that is used to compute the reference forces. In this way, the coverage of the conformational space is consistent with the level of theory of the reference forces, which is expected to lead to a better convergence for the equilibrium values in the bonded interactions (Eq. **??**). The disadvantage, however, is the limitation of the simulation time at the QM/MM level. At such a time scale it is unlikely to observe dihedral transitions in the QM subsystem. Therefore, the procedure described in this work provides a parameter set that can be safely used only for a specific conformation of the system. It is possible to generate a force field that also reproduces torsional barriers if the QM/MM dynamics are performed under the action of a bias potential that induces transitions in the available computational time. Possible alternatives could consist of extracting different configurations from a classical trajectory or to sample configurations along a given collective coordinate.

2.1.2 QM/MM Reference Forces

The target property to optimise the parameters in the bonded terms of the classical FF of the fragment included in the QM subsystem are the nuclear forces on the QM atoms computed at the reference configurations. If the QM/MM reference trajectory is performed based on Born-Oppenheimer (BO) MD, the reference forces can be stored during the reference MD run and no extra calculations have to be performed. We chose, however, to employ Car-Parrinello MD to generate the reference configurations. Along this trajectory the electronic structure is not exactly in the ground state and we can therefore not use the corresponding nuclear forces on the QM subsystem for the force matching. We compute BO reference forces with tight convergence criteria in an additional step on the subset of L reference configurations.

2.1.3 Fit of Atomic Point Charges

In our approach we aim to derive a set of charges that best reproduces the electrostatic properties of the QM subsystem. A particular innovation here is to fit the partial atomic charges directly from the QM/MM electrostatic potential according to a modified D-RESP scheme,[3,4] at difference to the conventional ESP[5,6] and RESP[7] procedures, exploiting more efficiently the rich information provided by the high-level reference calculations.[8]

We require the atomic charges to reproduce the electric field and potential on a grid that is defined by the positions of the *NN* atoms, i.e. all classical atoms that are explicitly coupled to the QM charge density. We include all reference configurations in the fitting. Furthermore, we employ weak restraints of the atomic charges to their respective Hirshfeld values.[9] These additional restraints are necessary in order to exclude chemically unreasonable and strongly conformation dependent solutions to the overdetermined problem.[3,7] The set of atomic charges $\{q_\alpha\}$ is optimised

by minimising the penalty function:

$$\chi^2(\{q_\alpha\}) = \sum_{l=1}^{L} \left[\sum_{\beta \in NN_l} \left(w^V \left(V_{\beta l}^{\mathrm{MM}} - V_{\beta l}^{\rho} \right)^2 + w^E \left| \mathbf{E}_{\beta l}^{\mathrm{MM}} - \mathbf{E}_{\beta l}^{\rho} \right|^2 \right) + \right.$$
$$\left. \sum_{\alpha \in \mathrm{QM}} w^H \left(q_\alpha - q_{\alpha l}^H \right)^2 \right] + w^Q \left(Q^{\mathrm{tot}} - \sum_{\alpha \in \mathrm{QM}} q_\alpha \right)^2 \tag{2}$$

The index l runs over all L reference configurations; β labels the classical atoms NN_l that are explicitly coupled to the quantum charge density in configuration l. α refers to atoms of the QM subsystem. The weighting factors w^V, w^E, w^H and w^Q can be tuned specific to each system in order to balance the individual contributions, as described in **??**. $V_{\beta l}^{\rho}$ and $\mathbf{E}_{\beta l}^{\rho}$ are the electrostatic potential and field, respectively, on the classical atom β in configuration l due to the QM charge density, while $V_{\beta l}^{\mathrm{MM}}$ and $\mathbf{E}_{\beta l}^{\mathrm{MM}}$ are the potential and field due to the classical point charges $\{q_\alpha\}$ that are being optimised. The third term in Eq. 2 restrains the charges $\{q_\alpha\}$ to their respective Hirshfeld values $q_{\alpha l}^H$ and the last term assures that the total charge of the set $\{q_\alpha\}$ matches the reference total charge of the QM electron density Q^{tot}. Minimising the penalty function χ^2 can be recast in the form of least-squares problem and solved as an overdetermined system of linear equations in $\{q_\alpha\}$.

2.1.4 Force Matching of the Bonded Interactions

The characteristic of our approach is to derive the parameters for the bonded interactions in a separate step from the non-bonded interactions. Once a new set of charges $\{q_\alpha\}$ has been determined, we therefore compute the classical non-bonded forces $\mathbf{F}_{l\alpha}^{\mathrm{MM_{nb}}}$ (for the QM atom α in configuration l) due to the electrostatics employing the new charges and the van der Waals interactions. In order to optimise the parameters in the bonded interactions we subtract the non-bonded forces from the QM/MM reference forces and minimise the penalty function under variations of the bonded parameters $\{\tau_{b_n}\}$

$$\sigma^2(\{\tau_{b_n}\}) = \sum_{l=1}^{L} \sum_{\alpha \in \mathrm{QM}} \left\| \mathbf{F}_{l\alpha}^{\mathrm{MM_b}} - \left(\mathbf{F}_{l\alpha}^{\mathrm{QM}} - \mathbf{F}_{l\alpha}^{\mathrm{MM_{nb}}} \right) \right\|^2 \tag{3}$$

2.2 Implementation in CPMD and Practical Remarks

We have implemented our QM/MM force matching protocol in the CPMD package with the release 3.15.[1] The first step in the QM/MM force matching procedure is the generation of the reference configurations. For the moment we have generated the reference configurations by extracting a given number of configurations with a given stride from a finite temperature QM/MM MD run performed within CPMD. However, since alternative schemes are possible we have left this open to the user and at the beginning of the QM/MM force matching run one only needs a set of configurations in a trajectory file in CPMD format. The format is also described in Section 3.

Once the reference configurations are generated the QM/MM force matching run can be launched. The force matching run can be controlled with a set of keywords in a separate block in the QMMM section of the CPMD input file. All keywords and options are described in the manual.

Like a normal QM/MM run, the force matching procedure needs a classical topology for the full system at the beginning. Since for performance reasons it is often preferred to run the classical simulations with other codes, appropriate converters are available to generate such an initial GROMOS topology.

The first step is the calculation of the QM/MM reference forces for the reference configurations that are read from the reference trajectory. The user can define a given stride to control the total number L of equally spaced configurations extracted from the reference trajectory. This represents the most time consuming part of the whole procedure and we have implemented a restart option in case a batch job system limits the calculation runtime below the need to compute all reference frames. During this run the QM/MM reference forces $\mathbf{F}_{l\alpha}^{\mathrm{QM}}$, the electrostatic potential $V_{\beta l}^{\rho}$, field $\mathbf{E}_{\beta l}^{\rho}$ and Hirshfeld charges $q_{\alpha l}^{H}$ are written to respective files.

The functional form of the non-bonded interactions implemented in CPMD is common to both Amber and GROMOS96 FFs:

$$E^{nb} = \sum_{\alpha < \beta} \left[\frac{A_{\alpha\beta}}{R_{\alpha\beta}^{12}} - \frac{B_{\alpha\beta}}{R_{\alpha\beta}^{6}} + \frac{q_{\alpha} q_{\beta}}{\varepsilon R_{\alpha\beta}} \right] \tag{4}$$

Standard exclusion rules for the electrostatic interactions are applied.[10] At the current state of the implementation the van der Waals parameters $A_{\alpha\beta}$ and $B_{\alpha\beta}$ are not optimised but are kept fixed to the original force field value. This choice is consistent with the QM/MM interaction Hamiltonian we use that retains the Lennard-Jones parameters from the classical force field.[9] Future developments will include a parameterisation protocol for the van der Waals parameters based on reference values obtained with the recently developed Dispersion Corrected Atom-Centered Potentials.[11–15] In the charge fitting procedure it is possible to specify chemically equivalent atoms which then will obtain the same charge values. Furthermore we implemented the capability to constrain individual charges to specific values. In order to solve the overdetermined system of linear equations associated to the minimisation of $\chi(\{q_\alpha\})^2$ with respect to $\{q_\alpha\}$ in Eq. 2 we have implemented an algorithm based on *QR* factorisation.[16]

In order to determine the optimal values for the weights w^V, w^E, w^H and w^Q in Eq. 2 the quality of the resulting charge set in reproducing the QM electrostatic properties can be measured for different values for the weights.[10] An example is given in the results section.

The bonded interactions can be calculated in CPMD using the AMBER functional form (Eq. **??**) or GROMOS96. The latter is written as

$$
\begin{aligned}
E^b = &\sum_{n=1}^{N^{\text{bonds}}} \frac{1}{4} K_{b_n} \left((b_n)^2 - (b_n^{\text{eq}})^2 \right)^2 + \sum_{n=1}^{N^{\text{angles}}} \frac{1}{2} K_{\theta_n} (\cos\theta_n - \cos\theta_n^{\text{eq}})^2 + \\
&\sum_{n=1}^{N^{\text{dihedrals}}} V_n [1 + \cos(\delta_n)\cos(m_n\phi_n)] + \sum_{n=1}^{N^{\text{impropers}}} \frac{1}{2} K_{\xi_n} (\xi_n - \xi_n^{\text{eq}})^2
\end{aligned}
\tag{5}
$$

where the first term represents the bond-stretching potential with force constant K_{b_n} and equilibrium value b_n^{eq}. The second term describes the angle bending potential with equilibrium value θ_n^{eq} and force constant K_{θ_n}. The third term represents the periodic dihedral potential with barrier height $2V_n$, the phase shift δ_n (0 or π) and the multiplicity m. And the last term assigns a harmonic potential to variations in the improper dihedral angles ξ_n.[1]

---

[1]Note that the Amber force fields employ a cosine form for improper torsions. Therefore, in case the force matching has been performed using the Gromos functional form, but at the end an Amber topology is needed, the improper dihedrals (harmonic form) have to be converted manually to the cosine form. Furthermore, the available Amber to

The minimisation of the penalty function in Eq. 3 with respect to the set of parameters $\{\tau_{b_n}\} = \left\{\{K_{b_n}\}, \{b_n^{eq}\}, \{K_{\theta_n}\}, \{\theta_n^{eq}\}, \{V_n\}, \{K_{\xi_n}\}\right\}$ can be formulated as a nonlinear least-squares problem. Since the derivatives of $\sigma^2(\{\tau_{b_n}\})$ with respect to the individual parameters τ_{b_n} are readily available we employed the gradient-based Levenberg-Marquardt algorithm as implemented in the MINPACK library.[17,18]

Special care has to be taken if the QM/MM boundary cuts a chemical bond. In order to avoid strong perturbations of the electronic structure the valence of the QM region has to be saturated. In CPMD this can be achieved by introducing a QM capping atom, typically a hydrogen atom. The capping atom is an artificial construction and its unphysical interaction with the MM environment is typically removed via exclusion rules. However, in the QM/MM force matching scheme further complications arise from the unphysical nuclear forces on the capping atom, since it is included in the QM region and therefore contributes to the QM/MM reference forces. We therefore suggest to employ the alternative scheme implemented in CPMD, in which the MM boundary atom is incorporated into the QM region and replaced by a specially parameterised monovalent pseudopotential.[?][2] This scheme has been applied successfully in QM/MM simulations over the past ten years. Note however, that the electron density in the surrounding of the monovalent pseudopotential is slightly perturbed and it is therefore expected that in our QM/MM force matching scheme the fitting of the charges leads to unreasonable solutions in this region. We found it best practice to fit, in a first step, atomic point charges for all atoms included in the QM region as described in Sec. 2.1.3. The charges on the monovalent pseudopotential and its nearest QM neighbours adopt chemically unreasonable values but they account for the perturbations of the electronic structure at the QM/MM boundary. The charges further away from the boundary are chemically sound and represent well the electrostatic properties of the surrounding of the QM region. In a second step, i.e. before fitting the bonded interactions, the charges on the monovalent pseudopotential and its nearest QM neighbours are replaced by their original FF values, while preserving the total charge

Gromos conversion tools treat the Amber improper torsions as regular dihedral interactions.

[2]In case reference configurations were already generated using a capping atom, the monovalent pseudopotential can simply be introduced at the point when the reference forces are re-calculated, however the capping atom would need to be removed at this point from the reference trajectory.

Q^{tot}. The final atomic point charges in this region are therefore not optimised but retained from the original FF, which should provide a more physically sound electrostatic representation here. Furthermore, it might be necessary to constrain the bond between the monovalent pseudopotential and the next QM atom during a QM/MM MD. A typical choice for the bond length is the equilibrium value in the original FF. After fitting the bonded interactions by force matching we recommend in such a case to retain all parameters involving one of these atoms at the original FF values.

This procedure to handle the parameters involved in interactions close to the QM/MM boundary works well if the QM region is chosen large enough and if the original classical point charges and bonded parameters in the vicinity of the monovalent pseudopotential are reasonably good. In the case of optimising parameters for the retinal moiety in rhodopsin, for example, we deliberately extended the QM region up to C_β of the LYS 296 side chain (see Fig. 1). Naturally, in case the QM/MM boundary does not involve any bonded interactions, e.g. a molecule in the QM region with the surrounding solvent treated at the classical level, no such complications arise.

The QM/MM force matching run in CPMD generates a modified GROMOS topology file. In case the classical simulations with the modified FF are to be run with a different code the user has to extract the respective parameters from this modified topology file and update the topology in the format of his choice by hand.
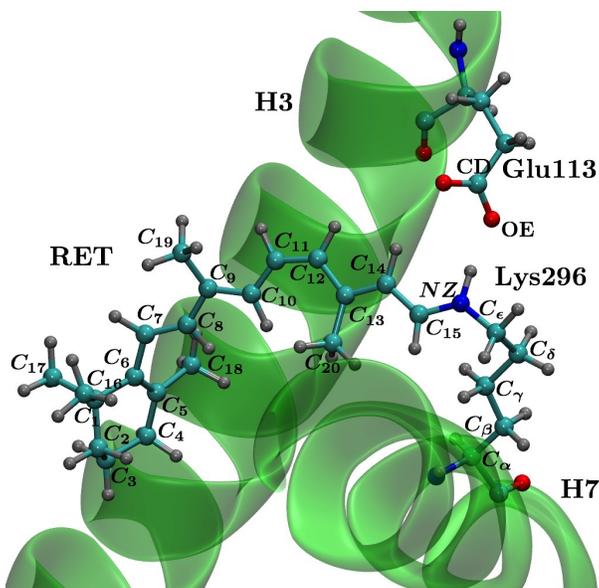
Figure 1: 11-cis retinal (RET) covalently bound by a protonated Schiff base (PSB) linkage to Lys296 (in helix H7) in ball and sticks representation. The counter ion Glu113 (in helix H3) is also shown.

Notes and References

(1) J. Hutter *et. al.*, Computer Code CPMD, version 3.15, IBM Corp. and MPI-FKF Stuttgart, 1990-2008, http://www.cpmd.org.

(2) Ercolessi, F.; Adams, J. *Europhys. Lett.* **1994**, *26*, 583–588.

(3) Laio, A.; VandeVondele, J.; Rothlisberger, U. *J. Phys. Chem. B* **2002**, *106*, 7300–7307.

(4) Laio, A.; Gervasio, F. L.; VandeVondele, J.; Sulpizi, M.; Rothlisberger, U. *J. Phys. Chem. B* **2004**, *108*, 7963–7968.

(5) Momany, F. A. *J. Phys. Chem.* **1978**, *82*, 592–601.

(6) Cox, S. R.; Williams, D. E. *J. Comput. Chem.* **1981**, *2*, 304–323.

(7) Bayly, C. I.; Cieplak, P.; Cornell, W.; Kollman, P. A. *J. Phys. Chem.* **1993**, *97*, 10269–10280.

(8) Bucher, D.; Guidoni, L.; Maurer, P.; Rothlisberger, U. *J. Chem. Theory Comput.* **2009**, *5*, 2173–2179.

(9) Laio, A.; VandeVondele, J.; Rothlisberger, U. *J. Chem. Phys.* **2002**, *116*, 6941–6947.

(10) Maurer, P.; Hugosson, H. W.; Colombo, M. C.; Rothlisberger, U. *J. Chem. Theory Comput.* **2007**, *3*, 628–639.

(11) Lilienfeld, O. A. v.; Tavernelli, I.; Rothlisberger, U. *Phys. Rev. Lett.* **2004**, *93*, 153004.

(12) Lin, I.; Coutinho-Neto, M.; Felsenheimer, C.; Lilienfeld, O. A. v. *Phys. Rev. B* **2007**, *75*, 1–5.

(13) Cascella, M.; Lin, I.-C.; Tavernelli, I.; Rothlisberger, U. *J. Chem. Theory Comput.* **2009**, *5*, 2930–2934.

(14) Aeberhard, P.; Arey, J.; Lin, I.; Rothlisberger, U. *J. Chem. Theory Comput.* **2009**, *5*, 23–28.

(15) Doemer, M.; Tavernelli, I.; Rothlisberger, U. *J. Chem. Theory Comput.* **2012**, *9*, 955–964.

(16) Lawson, C. L.; Hanson, R. J. *Englewood Cliffs* **1974**,

(17) Garbow, B. S.; Hillstrom, K. E.; More, J. J. *Implementation guide for MINPACK-1*; 1980.

(18) Moré, J. J. *Numerical analysis*; Springer Berlin Heidelberg, 1978; pp 105–116.

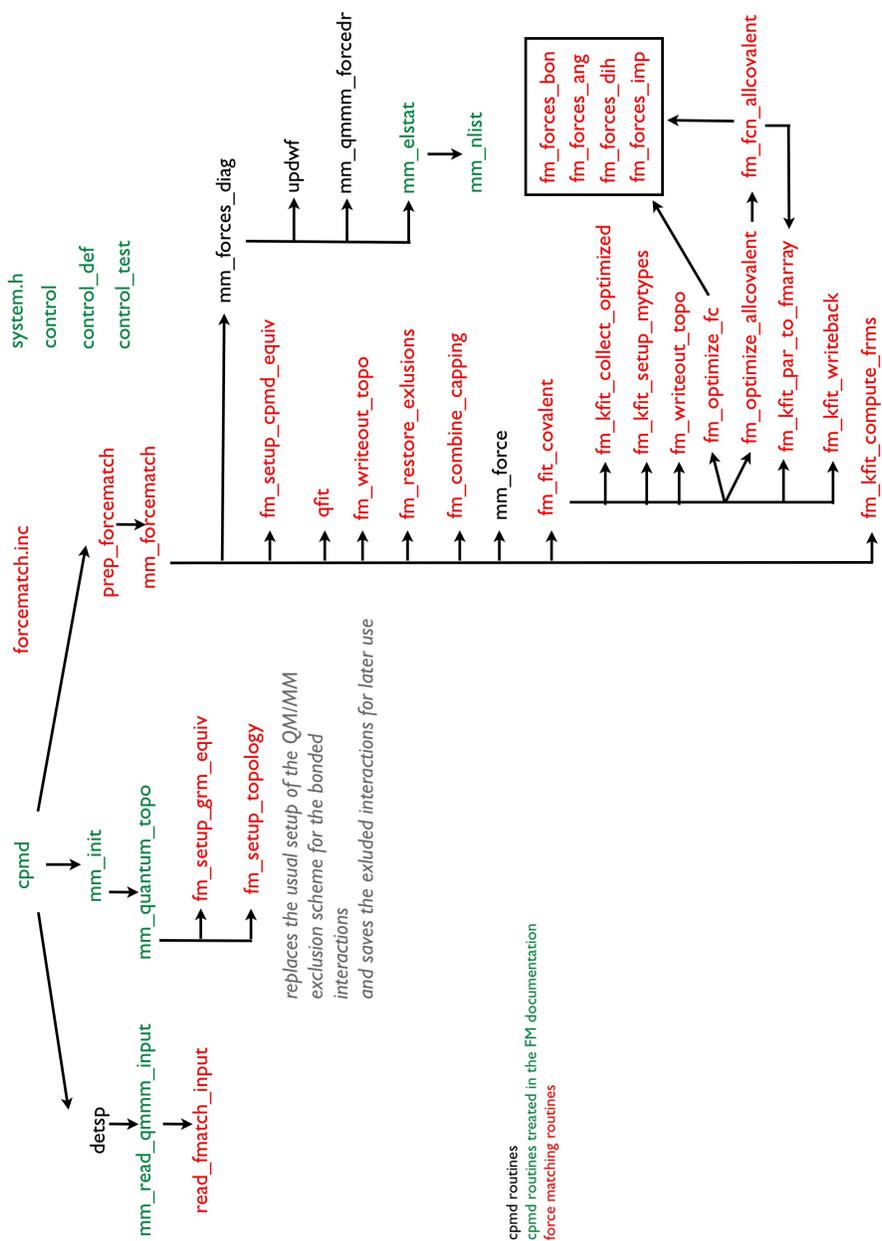# 3 Overview of the QM/MM Force Matching Routines in CPMD

Figure 2: flow chart of the force matching protocol in cpmd

13

4 Keywords

FORCEMATCH

The force matching keywords are read within the "&QMMM" section and have to be given within a block "[FORCEMATCH ... END FORCEMATCH]". Example:

```
FORCEMATCH
  READ REF TRAJ  FILE
   TRAJECTORY_REF
  READ REF STRIDE
   10
  WV
  1.0
  WF
  0.0
  WTOT
  1000000.0
  WQ GENERAL
  0.1
END FORCEMATCH
```

Figure 3: example of the forcematching block within the "&QMMM" section of a cpmd input file

READ REF FORCES [FILE],[COVALENT]

QM/MM reference forces are directly read from the file FM_REF_FORCES, i.e. no QM/MM SPs are computed. In case of a different file name with the option **FILE** it can be specified on the next line.

With the option **COVALENT** covalent forces are read from the file FM_REF_COVFORCES.

READ REF TRAJ [FILE]

Read reference trajectory from file TRAJECTORY_REF (or set the **FILE** option to read a non-default file name from the next line) with a given stride and compute single points on the respective frames.

RESTART SP

In cases that in a previous run not all of the SPs could be computed (e.g. limited wall time) this flag indicates cpmd to restart the SP calculations. The FM_REF* files from the previous run have to be present and they will be appended. Check that the frames contained in the already existing FM_REF* files are consistent. There is a consistency-check, but you never know...

READ REF STRIDE

Stride to apply when reading the FM_REF_FORCES file is read from the next line. Default=1, i.e. every frame is used for the SP calculations.

TOPOL OUT

Filename for the final topology file. Default=FMATCH.top.

INITWF [OFF]

Generate an initial guess for the wfkt for the SP calculations based on AOs (default). With the **OFF** option the wfkt of the previous frame is used as an intial guess.

CHARGES [ONLY,NO],[FIX]

Charge fitting is on by default and can be switched off with the **NO** option. In this case the charges from the initial topology will not be modified. **ONLY** will let the program stop after the charge fitting and the other parameters are not updated.

With the **FIX** option target values for the restraints in the charge fitting on specific atoms can be specified by the user. Usually the charges are restraint to the respective Hirschfeld values (see equation xx). On the next line the number of charges to be fixed has to be given and then the corresponding number of lines with: gromos index   charge

OPT QFIT WEIGHT

[NOT IMPLEMENTED YET] optimize the weights (variables yy in equation xx) in the charge fitting procedure by a build-in automatic procedure.

WV

Weight on the potential in the charge fitting (equation xx). Default=0.1.

WF

Weight on the field in the charge fitting (equation xx). Default=0.0.

WQ INDIVIDUAL

Weights on the charge restraints (variable in equation xx) can be given individually here. From the next line the total number of individual weights is read. Then the lines with: gromos index weight.

WQ GENERAL

The weight for all the charge restraints that were not specified by individual weights can be given on the next line. Default=0.1

WTOT

Weight of the total charge contribution in the charge fitting (equation xx). Default=1.0E7

EQUIV

Specify equivalent atoms. Syntax:

```
    EQUIV

n_equiv

atom1    atom4

atom1    atom3

...

atom5    atom7
```

There are n_equiv equivalencies specified (n_equiv lines are read from the input). For each pair of equivalencies the gromos indexes have to be specified on one separate line. The lower index has to be given first!

If an atom is equivalent to more then one other atom. E.g. atom1,atom3 and atom4 are equivalent. Then this has to be encoded by:

```
    atom1    atom3

atom1    atom4
```

and not by:

```
    atom1    atom4

atom3    atom4
```

where atom1 has a lower gromos index then atom3 and atom3 has a lower one then atom4. Per default no equivalencies are assumed.

OPT FC ONLY

Serves as a flag to remove the equilibrium values of the bonded interactions from the list of fitted parameters. I. e. only force constants are fitted for the bonded interactions.

NO BONDS

Do not fit bonds. Default=.false.

NO ANGLES

Do not fit angles. Default=.false.

NO DIHEDRALS

Do not fit dihedrals. Default=.false.

NO IMPROPERS

Do not fit improper dihedrals. Default=.false.

FMCAP

Signal here whether hydrogen capping is used for the QM/MM coupling. Give the number of capping hydrogens on the next line and then one line for each dummy hydrogen with its gromos index and the index of the capped atom wo which the respective dummy H is linked to Format:

```
   FMCAP
fm_n_cap
ind_dummy_H1    ind_capped_C1
...
```

MAXITER

Give on the next line the maximal number of iterations for the non-linear fitting procedure of the bonded interactions. Default=500

COMPUTE RMS [NO]

Per default the RMS on the forces is computed after the fitting has been completed. Switch it off with the **NO** option.

5   Files generated

Some of the following default filenames can be changed via the respective keywords in the "FORCE-MATCH" block.

TRAJECTORY_REF

Reference trajectory. Same format as the cpmd "TRAJECTORY" file. This file is read with a given stride and QM/MM single points are computed for the selected frames. "NEW DATA" marks can be handled by the program.

FM_REF_CHJ

Hirschfeld charges on the QM atoms for all the frames that were selected for the SP.Format: Two lines per frame. First line contains frame index from the original "TRAJECTORY_REF" file. Second line gives the Hirschfeld charges on the QM atoms in cpmd ordering.

FM_REF_PIP

For each frame extracted from the "TRAJECTORY_REF" file a line is printed with the number of NN atoms in this very frame and the original frame index. Then, for all the QM and NN atoms (in

Gromos order) in this frame a line is printed with the columns:

1-3 position *x,y,z*

4 label "QM" or "MM"

5 classical charge

6 electrostatic potential due to classical charges

7 Rsmear

8 electrostatic potential due to QM nuclei

9 electrostatic potential due to QM electronic density

10-12 electrostatic field *x,y,z*

FM_REF_FORCES

For each frame extracted from the reference trajectory and for which a QM/MM SP was calculated, the QM/MM forces on the QM atoms are dumped into this file. One line per frame with the original frame index and the number of QM atoms. Then for each QM atom in cpmd ordering:
atom index,x,y,z,fx,fy,fz

FM_REF_COVFORCES

QM/MM reference forces on the QM atoms minus classical non-bonded forces. Forces to fit the bonded interactions.

QFIT.top

Updated topology file after the charge fitting. The bonded parameters in here have no meaning, since they reflect the modifications needed for the QM/MM coupling (exclusions!).

FMATCH.top

Updated topology file after the force matching.

6 CPMD routines

system.h

MLOGO increase and logical FMATCH as primary flag.

control

Search CPMD section for FORCEMATCH keyword.

control_test

Check for mutually excluding job keywords.

control_def

FMATCH=.false.

mm_init

If forcematching call mm_quantum_topo even for a classical run.

MM_Interface/mm_nlist

Some modification on the atoms to include in the NN list.

MM_Interface/mm_elstat

Write the FM_REF_PIP and FM_REF_CHJ files.

mm_read_qmmm_input

`mm_read_qmmm_input.F`

- default values for many of the forcematching parameters are set

- call `read_fmatch_input`

mm_quantum_topo

`mm_quantum_topo.F`

Classical topology is modified to fit to the QM/MM coupling. For the force matching we need something more sophisticated for the bonded exclusions and the initial purely MM exclusions have to be saved before they are modified.

    identify quantum atoms: mm_q

    non-bonded interactions: mm_q_nb

    if FMATCH **then**

        call fm_setup_grm_equiv

        call fm_setup_topology

    else

        set up the usual QM/MM exclusions for the bonded interactions

        create a new bond type with force constant=0.0

        go trough all the bonds in the topology

        if they are QM-QM bonds, then change their type to the new bond type

        the same with angles, dihedrals and improper dihedrals

    end if

    if FMATCH **then**

        non-bonded exclusions are saved for later use

        INE(I): last index in the exclusion list of atom I

KNE(I): first index in the exclusion list of atom I

JSNE: non-bonded exclusions. Exclusions J of atom I are positioned at

JSNE(KNE(I)+1),...JSNE(KNE(I)+INE(I)). all J must be > I and in ascending order.

INE14

KNE14

JSNE14: 1-4 exclusions. Organization of the arrays analogous to JSNE.

end if

non-bonded exclusions are modified to fit the QM-MM coupling

7 Force Matching Routines in CPMD

prep_forcematch

fm_setup_grm_equiv

`forcematch_util.F`

Setup equivalencies in gromos ordering, i.e. fill the array grm_equiv.

From the input

```
   EQUIV
3    4
3    5
6    8
```

we have: fm_equiv(1,1)=3, fm_equiv(2,1)=4, fm_equiv(1,2)=3, fm_equiv(2,2)=5, fm_equiv(1,3)=6, fm_equiv(2,3)=8

grm_equiv$(i) = i \quad \forall \, i = 1, ..., $nr solute atoms {Initially make any atom equivalent to itself}

from the input we have fm_equiv()

update grm_equiv(:) according to the equivalencies in the input: atom i is equivalent to atom grm_equiv(i)

and we get: grm_equiv $= [1, 2, 3, 3, 3, 6, 7, 6, 9, 10, ..., \text{NRPT}]$ (NRPT: total number of solute atoms)

fm_setup_topology

`forcematch_kfit.F`

1. Create new bonded interaction (bonds, angles, dihedrals, improper dihedrals) types if necessary: increase NBTY, NTTY, NQTY, NPTY

2. Determine number of bonded interaction types to optimize: fm_nbonopt, fm_nangopt, fm_nimpopt, fm_ndihopt

3. Create lists of the indexes for the bonded interaction types to optimize: fm_bonopt(1:fm_nbonopt), fm_angopt(1:fm_nangopt), fm_impopt(1:fm_nimpopt), fm_dihopt(1:fm_ndihopt)

4. set to zero all the force constants of all the QM-only bonded interactions: CB, CT, CQ and CP. Store the values for later: fm_kbon, fm_kang, fm_kimp, fm_kdih.

for all bonds i in the topo involving hydrogens **do**

  if bond i is QM-QM **then**

    for all other bonds j **do**

      if type(bond j) = type(bond i) **and** bond j is QM-MM or MM-MM **then**

        create a new type: NBTY $\leftarrow$ NBTY+1

        type(bond i) $\leftarrow$ NBTY

        copy old parameters: CB(NBTY) $\leftarrow$ CB(old_type), B0(NBTY) $\leftarrow$ B0(old_type)

        flag type(bond i) to be optimized

      end if

    end for

if no new type was created **and** type(bond i) was already flagged to be optimized **then**

{there is already an earlier QM-QM bond of this type}

   for all bonds $j < i$ **do**

      if type(bond j) = type(bond i) **and** bonds i,j are not equivalent **then**

         create a new type: NBTY $\leftarrow$ NBTY+1

         type(bond i) $\leftarrow$ NBTY

         copy old parameters: CB(NBTY) $\leftarrow$ CB(old_type), B0(NBTY) $\leftarrow$ B0(old_type)

         flag type(bond i) to be optimized

      end if[if they are equivalent we don't need a new type]

   end for

end if

for all other bonds $j > i$ **do**

   if bond j is equivalent to bond i **then**

      type(bond j) $\leftarrow$ type(bond i)

   end if

end for

end if[the QM-MM and MM-MM bonds we do not want to touch]

end for

the same is happening for all bonds without hydrogens

fm_nbonopt is calculated

fm_bonopt is filled with the indexes of the bonds to be optimized

the same with the angles involving hydrogens

the same is happening for all angles without hydrogens

fm_nangopt

fm_angopt

improper dihedrals involving hydrogens

improper dihedrals not involving hydrogens

25

fm_nimpopt

fm_impopt

proper dihedrals involving hydrogens. If there are two or more dihedrals involving QM atoms only, being of the same type and being equivalent: No new types will be created. Dihedrals that already have force constants=0.0 are not flagged to be fitted.

proper dihedrals not involving hydrogens

fm_ndihopt

fm_dihopt

save force constants of QM-QM bonds for later and set them to zero:

$\quad$ fm_kbon(i) $\leftarrow$ CB(fm_bonopt(i)), CB(fm_bonopt(i)) $\leftarrow 0.0 \ \forall \ i = 1, ..., $fm_nbonopt

$\quad$ fm_kang(i) $\leftarrow$ CT(fm_angopt(i)), CT(fm_angopt(i)) $\leftarrow 0.0 \ \forall \ i = 1, ..., $fm_nangopt

$\quad$ fm_kimp(i) $\leftarrow$ CQ(fm_impopt(i)), CQ(fm_impopt(i)) $\leftarrow 0.0 \ \forall \ i = 1, ..., $fm_nimpopt

$\quad$ fm_kdih(i) $\leftarrow$ CP(fm_dihopt(i)), CP(fm_dihopt(i)) $\leftarrow 0.0 \ \forall \ i = 1, ..., $fm_ndihopt

fm_fit_covalent

forcematch_kfit.F

This is the main driver routine to fit the parameters of the bonded interactions: Force constants, equilibrium bond lengths, angles, improper dihedrals and dihedrals.

Atomic units are used!

If only force constants are fitted (keyword: OPT FC ONLY) the problem can be cast in the form of an overdetermined system of linear equations, which is solved with a lapack routine. If both force constants and equilibrium values are fitted (default) the equations are non-linear and a minpack routine is used to solve it

calculate total number of parameters: npar

collect all the bonded interactions that are concerned by the parameters involved in the optimization: call fm_kfit_collect_optimized and call fm_kfit_setup_mytypes

if force constants only **then**

call fm_optimize_fc

else

generate an initial guess: copy the arrays fm_kb, fm_b0, fm_kt, fm_t0, fm_kq, fm_kp (which, in the beginning hold the values from the initial topology) to the array par(1:npar)

call fm_optimize_allcovalent. Driver routine that handles the routines to set up and solve the overdetermined system of non-linear equations

end if

call fm_kfit_par_to_fmarray. Write the optimized parameters from the array par back to fm_kb, ...

call fm_kfit_writeback. Copy the optimized parameters from the fm_$*$ arrays to the gromos arrays BO, CB,...

fm_kfit_collect_optimized

`forcematch_kfit.F`

After having identified all the bonded interaction types that are going to be optimized in fm_setup_topology, all the indexes involved in these interactions are assembled here.

1. remember fm_nbonopt: total number of bond types to optimized

2. total number of bonds being optimzed: fm_nbon

3. for all the bonds i=1,...,fm_nbon that are being optimized:

   - fm_ib(0,i) holds the fm bond type index. Gromos bond type index: fm_bonopt(fm_ib(0,i))

   - fm_ib(1,i) and fm_ib(1,i) hold the cpmd indexes of the atoms involved in the bond

fm_kfit_setup_mytypes

`forcematch_kfit.F`

1. The parameters that were set to zero in fm_setup_topology are now put back in place: CB, CT, CQ, CP

2. convert the gromos parameters to a.u. and fill the corresponding fm arrays: fm_kb , fm_b0, fm_kt, fm_t0, fm_kq, fm_q0, fm_p0, fm_pn, fm_kp.

   E.g. $\mathrm{fm\_kb}(i) = \mathrm{kb\_grm2au} \cdot \mathrm{CB}(\mathrm{fm\_bonopt}(i)) \ \forall \ i = 1, ..., \mathrm{fn\_nbonopt}$

fm_optimize_allcovalent

`forcematch_kfit.F`

Optimizes the parameters of all bonded interactions. This is an overdetermined non-linear problem and the routine lmder1 from minpack is used to solve it.

lmder1 calls fm_fcn_allcovalent as an external routine to set up the matrix problem.

success and error messages are printed.

fm_fcn_allcovalent

`forcematch_kfit.F`

Service routine for lmder1 (modified Levenberg-Marquardt least square minimization for an overde-termined system) to build up the jacobian matrix (jac) and the target vector (fdiff) in each iteration. The rough structure is prescribed by lmder1. For further information on the optimization proce-dure consult the lmder1 documentation.

In the beginning the the fm_$*$ arrays are updated. Then, depending on the value of iflag, either the jacobian or the fdiff array is updated.

The non-linear problem: fjac (jacobian) is a $(3 * \mathrm{nr_{qm}} * \mathrm{nframes} \times \mathrm{npar})$ matrix. par is the array of size npar with the parameters being optimized. fdiff is the array of size $3 * \mathrm{nr_{qm}} * \mathrm{nframes}$ with the function values to be minimized.

$$\mathrm{fjac} * \mathrm{par} = \mathrm{fdiff} \tag{6}$$

for one frame i, jac^1 is a $(3 * \text{nr}_{\text{qm}} \times \text{npar})$ matrix:

$$\text{jac}_i * \text{par} = \text{fvec}_i \tag{7}$$

where

$$\text{fjac} = \begin{pmatrix} \text{jac}^{(1)} \\ \text{jac}^{(2)} \\ \vdots \\ \text{jac}^{(\text{nframes})} \end{pmatrix} \tag{8}$$

$$\text{jac}^{(i)} = \begin{pmatrix} \frac{\partial}{\partial \text{par}(1)}\left(\frac{\partial E^i}{\partial R^i(1)}\right) & \frac{\partial}{\partial \text{par}(2)}\left(\frac{\partial E^i}{\partial R^i(1)}\right) & \cdots & \frac{\partial}{\partial \text{par}(\text{npar})}\left(\frac{\partial E^i}{\partial R^i(1)}\right) \\ \frac{\partial}{\partial \text{par}(1)}\left(\frac{\partial E^i}{\partial R^i(2)}\right) & \frac{\partial}{\partial \text{par}(2)}\left(\frac{\partial E^i}{\partial R^i(2)}\right) & \cdots & \frac{\partial}{\partial \text{par}(\text{npar})}\left(\frac{\partial E^i}{\partial R^i(2)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial \text{par}(1)}\left(\frac{\partial E^i}{\partial R^i(3*\text{nr}_{\text{qm}})}\right) & \cdots & \cdots & \frac{\partial}{\partial \text{par}(\text{npar})}\left(\frac{\partial E^i}{\partial R^i(3*\text{nr}_{\text{qm}})}\right) \end{pmatrix} \tag{9}$$

where E^i is the energy corresponding to the configuration (frame) $\overrightarrow{\mathbf{R}}^i$, which in turn depends on the parameter set par. The problem is non-linear, because of the structure of the energy expression the elements of the jacobian and the fvec depend on some of the other parameters:

$$\text{jac}^{(i)}_{mn} = \frac{\partial}{\partial \text{par}(n)}\left(\frac{\partial E_i(\{\text{par}\})}{\partial R_m}\right) \quad \Rightarrow \quad \text{jac}^{(i)}_{mn}(\{\text{par}(l)\}, l \neq n) \tag{10}$$

The target vector holds the difference forces from the QM/MM reference (minus non-bonded interactions):

$$\text{fdiff} = \begin{pmatrix} \text{fvec}^{(1)} \\ \text{fvec}^{(2)} \\ \vdots \\ \text{fvec}^{(\text{nframes})} \end{pmatrix} \tag{11}$$

$$\text{fvec}^{(i)}_m = \frac{\partial E(\{\text{par}\})}{\partial R^i(m)} - \frac{\partial E_{\text{ref, cov}}}{\partial R^i(m)} \tag{12}$$

29

The parameter array is organized as follows:

$$par = \big[fm\_kb(1), fm\_kb(2), \ldots, fm\_kb(nbonopt), fm\_b0(1), \ldots, fm\_b0(nbonopt), fm\_kt(1), \ldots, fm\_kt(nangopt),$$

$$fm\_t0(1), \ldots, fm\_t0(nangopt), fm\_kq(1), \ldots, fm\_kq(nimpopt), fm\_kp(1), \ldots, fm\_kp(ndihopt)\big]$$

The routine can be called with two values for the flag iflag.

iflag=1: update the fdiff array without changing the jacobian. In this case all the fm_forces_$*$ routines are called with the flag 'ff', i.e., the forces are computed.

iflag=2: update the jacobian. All the fm_forces_$*$ routines are called with the flag 'dk' and, where applicable, 'd0' in order to compute the derivative of the forces with respect to the respective parameters.

fm_kfit_par_to_fmarray

`forcematch_kfit.F`

fm_optimize_allcovalent and fm_optimize_fc put the results into the array par(1:npar). Here the arrays fm_kb, fm_b0, fm_kt, fm_t0, fm_kq, fm_kp are updated with the optimized parameters. In addition, fm_fcn_allcovalent needs this in each iteration to build up a new jacobian matrix.

fm_kfit_compute_frms

`forcematch_kfit.F`

Compute RMS between the QM/MM and the new classical forces on the atoms involved in the interactions that were fitted.

read_fmatch_input

`mm_read_qmmm_input.F`

Reads the FORCEMATCH block in the "&QMMM" section and sets the respective parameters.

forcematch.inc

forcematch.inc

Include file for the forcematching subroutines.