

On the reparameterization of torsional potentials in molecular mechanics:

A tutorial based on NAMD code and CHARMM force field

Torsions???

Among the terms contributing to the description of the molecular shape in a molecular mechanics (MM) force field (FF), the ones describing the torsional potentials are probably the most important and the most indicated for a careful check before starting a simulation campaign. With torsional potential, it is meant the variation of the energy as a function of the torsional (or dihedral) angle, by definition the angle between four atoms connected in a row 1-2-3-4, which can span a full turn. In mathematical terms, the dihedral angle φ is a pseudoscalar, i. e. a quantity that behaves like a scalar, except that it changes sign under a parity inversion (a flip of the sign of one coordinate axis) such as improper rotations, while a true scalar does not [Note1]. In its $[-\pi;+\pi]$ definition, it can be calculated from the atomic coordinates $\vec{r}_1, \vec{r}_2, \vec{r}_3, \vec{r}_4$ and their differences $\vec{r}_{12} = \vec{r}_2 - \vec{r}_1, \dots$, as (see Appendix A for the corresponding Fortran code):

$$\varphi = \cos^{-1} \left[\frac{(\vec{r}_{12} \times \vec{r}_{23}) \cdot (\vec{r}_{23} \times \vec{r}_{34})}{|\vec{r}_{12} \times \vec{r}_{23}| \cdot |\vec{r}_{23} \times \vec{r}_{34}|} \right], \quad (1)$$

with its sign given by

$$\text{sgn}(\varphi) = \text{sgn} \{ [(\vec{r}_{12} \times \vec{r}_{23}) \times (\vec{r}_{23} \times \vec{r}_{34})] \cdot \vec{r}_{23} \}. \quad (2)$$

The origin of the importance of torsional potential can be found in the low energy associated by the change of a dihedral angle (with the exception of dihedrals between atoms belonging to the same aromatic ring, see figure 1, we are speaking of energy barriers of few kcal/mol), or in other words the in “softness” of torsional vibrational modes. The consequences are multiple, among them:

- these modes can be strongly anharmonic, hence they are described in the FF with a series of cosines (see eq. 1), and not with parabolas like bending and stretching terms
- the torsional potential has several local minima separated by small energy barriers
- the parameters describing the potential are difficult to transfer to a molecule to another
- the variation of the conformation (the value of the local minimum effectively populated) strongly affects the shape of the molecule
- the conformation or the population of conformations may quickly change with temperature and from a phase to another (see e.g. [Pizzirusso 2011]).

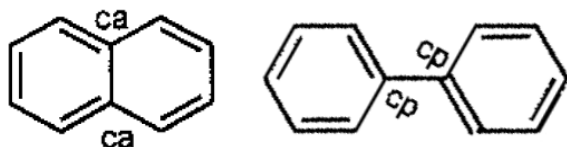


Figure 1: Example of atom types. In naphthalene (left), all the carbons belong to the same type “CA”, while for biphenyl a new type “cp” must be introduced for bridge carbons, as they experience a different torsional potential with respect to the carbons belonging to the same phenyl ring [Wang 2004].

In standard FFs, the torsional parameters are specified for each quadruple of atom types, where the type identifies elements sharing the same chemical environment, see reference [Wang 2004] and figure 1.

In the simplest case, only the type of the two central atoms (D_1 and D_2 in figure 2) is specified, with parameters that disregard the type of the two extreme atoms, indicated with a wildcard (“X”). It is worth noting that the total energy expression of the CHARMM force field reported below (equation 3), the sum over the dihedrals runs on

every quadruple of atom types, and that, depending on

the implementation, the number of cosines in the Fourier series is often limited to a maximum of six terms for each of them, but with freedom in the six values of n chosen.

$$\begin{aligned}
U_{total} = & \sum_{bonds} K_b (r - r_{eq})^2 + \\
& \sum_{angles} K_a (\theta - \theta_{eq})^2 + \\
& \sum_{dihedrals} \sum_{n=0}^N V_n [1 + \cos(n\varphi + \gamma_n)] + \sum_{atoms} \sum^{i<j} \left\{ \sqrt{\epsilon_i \epsilon_j} \left[\left(\frac{\sigma_i + \sigma_j}{2r_{ij}} \right)^{12} - \left(\frac{\sigma_i + \sigma_j}{2r_{ij}} \right)^6 \right] + \frac{q_i q_j}{r_{ij}} \right\}.
\end{aligned}
\tag{3}$$

To complete the picture we should keep in mind that the long range electrostatic and van der Waals contributions (described with the sum over atoms in equation 3), are normally excluded for atoms separated by two bonds (“1-3 interactions”) and are scaled by a given factor for atoms separated by three bonds (“1-4 interactions”). Clearly these 1-3 and 1-4 interactions are relevant in the case of dihedral potentials. 1-4 factors of 1/2 and 5/6 are used for vdW and electrostatics in the most adopted OPLS-AMBER FF, respectively [Jorgensen 1996]. To be more specific we take the sketch of figure 2 as example. In that case, only one torsion is present, involving all the quadruples of atoms A_i - D_1 - D_2 - B_i . The number of quadruples for each torsion is sometimes called the *multiplicity* of the dihedral, from herewith called N_Q . This number is given by multiplying the number of atoms connected to D_1 and D_2 . In the example, $N_Q = N_A \times N_B = 9$, consisting in the quadruples (A_1 - D_1 - D_2 - B_1), (A_2 - D_1 - D_2 - B_1), (A_3 - D_1 - D_2 - B_1), (A_1 - D_1 - D_2 - B_2), (A_2 - D_1 - D_2 - B_2), (A_3 - D_1 - D_2 - B_2), (A_1 - D_1 - D_2 - B_3), (A_2 - D_1 - D_2 - B_3), (A_3 - D_1 - D_2 - B_3). For each of these quadruples the corresponding parameters must be present in the dihedral section of the parameter file [Note2]. For instance if the atom type of atoms A_i , D_1 , D_2 , B_i are named $A1$, $D1$, $D2$, $B1$, we could have:

```

DIHEDRAL
X  D1  D2  X    2.1  3   180.0
X  D1  D2  X    0.2  6   180.0
A1 D1  D2  B3   2.7  4    0.0

```

Starting with the format, after the four atom type labels we find the constant V_n (in kcal/mol), n (the multiplicity of the cosine) and the phase γ_n . In this example, two cosine terms with $n=3$ and 6 are used for all the quadruples but (A_1 - D_1 - D_2 - B_3) where the explicit term with $n=4$ overrides the previous definitions. There are neither limitation in the integer values n can assume, nor in the phase angle. A negative sign of V_n is equivalent to a phase of 180°. For instance for the CF_3 - CF_2 C- torsion of 1, 1, 1, 2, 2-pentafluorobutane (figure 2) we used [Rosconi 2012]:

```

F  CF  CF  F    0.156  3  0.0
F  CF  CF  CF   0.156  3  0.0
CF  CF  CF  CF  -0.2531 1  0.0
CF  CF  CF  CF  -0.2002 2  0.0
CF  CF  CF  CF  -0.9107 3  0.0
CF  CF  CF  CF   0.5057 4  0.0
CF  CF  CF  CF   0.0986 5  0.0
CF  CF  CF  CF  -0.1229 7  0.0

```

As already mentioned, the existing parameterizations for dihedrals are often not too accurate (see e.g. [Berardi 2005]) and it is good practice to calculate a more reliable potential with quantum chemistry methods and re-parameterize the FF. This is particularly relevant for torsions involving rigid aromatic fragments or heteroatoms. The choice of the level of calculation depends on the size of the molecule: for small ones (max 30 atoms?), the “golden standard” is CCSD(T) at the extrapolated complete basis set limit [Riley 2012]; as a matter of fact, we never tried this method, mainly because the molecules we study have normally around 100 atoms or more. In this case, and in presence of many torsions to scan, CCSD(T) is prohibitive, and usually we resort to long range corrected Density Functional Theory (DFT) functionals, such as B97D [Grimme 2006] or wB97XD [Chai 2008], possibly with a triple zeta basis set (cc-pVTZ). However B3LYP/cc-pVDZ or /6-31G* [Becke

1993] is still the reference method in the literature and well accepted by referees, although in the family of hybrid DFT/Hartree Fock functionals the double hybrid B2PLYP(-D) [Schwabe 2007] is considered the best method [Sancho-Garcia 2009], albeit quite expensive in its Gaussian09 implementation. Ideally, all the calculations should be run with a triple zeta basis set with polarization and diffuse functions (e.g. aug-cc-pVTZ or 6-311+G*, 6-311++G**). In practice, it's a matter of choosing the best compromise between accuracy, feasibility, and deadlines/ time available for completing the force field setup.

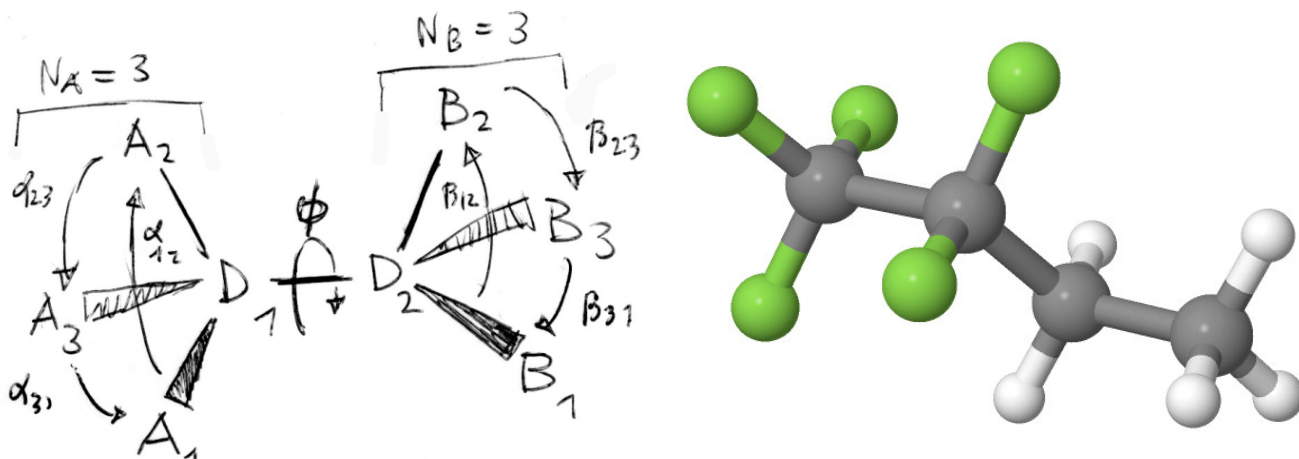


Figure 2: Left: scheme of the dihedral angle $A_1-D_1-D_2-B_1$; right: a real molecule featuring three types of dihedrals (one for each carbon-carbon bond): 1,1,1,2,2-pentafluorobutane.

To be more precise, the total torsional energy potential $-U(\varphi)$ from here - should be calculated with constrained energy minimizations: for each value of φ , approximated in discrete steps of 5-30°, a constrained energy minimization must be run, letting the other molecular degrees of freedom relaxed. In the FF, $U(\varphi)$ corresponds to the sum of all the terms corresponding to the quadruples i of atoms centered at the torsion, $u_i(\varphi)$, **opportunistically phase shifted** – in fact the definition of φ is not consistent for all the quadruples. In the figure 2 example, if we define φ as the dihedral angle between the atoms $A_1-D_1-D_2-B_1$, than for at the same geometry the dihedral for $A_1-D_1-D_2-B_2$ will be $\varphi + \beta_{12}$, and $A_2-D_1-D_2-B_3$ will be $\varphi + \beta_{12}$. So in general a quadruple i will be out of phase of an angle δ_i which value depends on the quadruple chosen for the definition of φ :

$$U(\varphi) = \sum_{i=1}^{N_Q} u_i(\varphi + \delta_i) = \sum_{i=1}^{N_Q} \sum_{n_i} V_{n,i} [1 + \cos(n_i\varphi + n_i\delta_i + \gamma_{n,i})] \quad (4)$$

On one hand this definition allows for a great flexibility in defining the FF parameters; on the other, multiple choices are possible for reproducing the torsional potential, depending on the types of the atoms linked to D_1 and D_2 . In the simplest and luckier cases, symmetry or local symmetry (the torsional potential is affected only by atoms about 5 Å away from D_1 and D_2) helps us in the definition of the better function for fitting $U(\varphi)$: the δ_i angles are all equal, the atom types A_i and B_i are all the same and $U(\varphi)$ reflects the symmetry of the chemical environment. In those case, experience and mathematics tells us that the choice of n_i respecting this symmetry helps and quicken the derivation of suitable parameters, although other solutions can be found. For instance for the Cp-Cp dihedral in figure 1 only even cosine terms respect the symmetry and thus odd terms should be excluded from the “basis set”. In asymmetric cases, it is more convenient to adopt a symmetric potential for all atom types quadruples but one, and to “charge” the asymmetry only on one quadruple type, possibly the one with lowest occurrence – see the CF_3-CF_2C- example below.

And so what should I do?

Let's take a practical example, and suppose that we have calculated the "correct" torsional potential for dithiophene (T2) with the help of some quantum chemistry calculations, like in figure 3. First of all, also the quantum chemistry potential $U^{QM}(\varphi)$ can be conveniently fitted and expressed with a series of cosines – very effectively for ground state calculations, a bit less for charged molecules, in particular if DFT is used (figure 3). It could be tempting to transfer directly the fit parameters to the force field, however this is not possible: first because all the terms in equation 3 contribute to the effective torsional potential, and in particular the van der Waals and electrostatic intermolecular interaction, but also the stretching of the bond between D_1 and D_2 . We call these "external" contributions $U^E(\varphi)$ while we keep the symbol $U(\varphi)$ for the explicit force field torsional term shown in equation 4. Secondly, depending on the symmetry of the atoms involved, it is not straightforward to translate the $U^{QM}(\varphi)$ profile into the sums on the right hand side of equation 4, although in the ideal case proper symmetry, $U^E(\varphi) = 0$ and only one atom type for the A_i and B_i , $U(\varphi) = N_Q u_i(\varphi)$ could hold. In the most common case it does, but still $U^E(\varphi) \neq 0$.

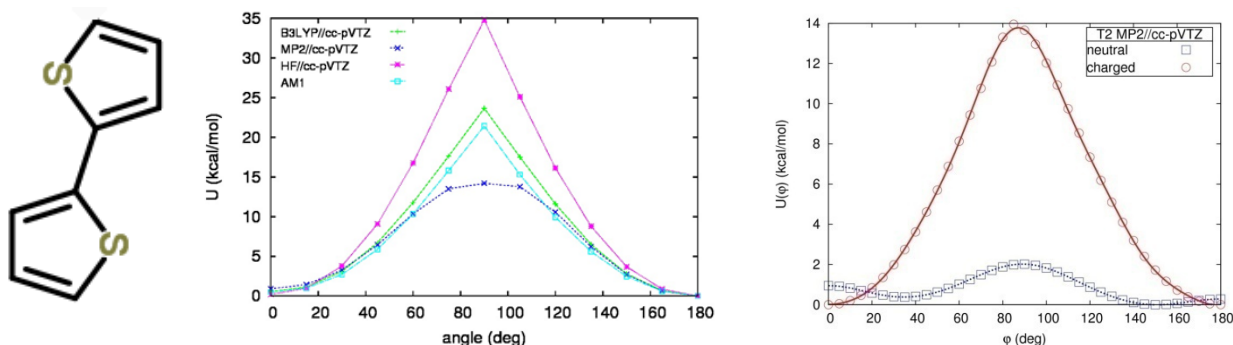


Figure 3: From the left, dithiophene (T2, C8S2H6) chemical structure, T-T torsional potential calculated with different methods for the positively charged molecule, the same potential calculated at MP2//cc-pVTZ level for positively charged and neutral state (points) and corresponding fit with a series of cosines (continuous lines). As the definition of the angle is not unique, it must always be specified: in this case it the S-C-C-S quadruple has been selected.

The problem of deriving the FF parameters than translates into finding a way for evaluating the function $U^E(\varphi)$ produced by the FF. Known this, the explicit FF parameters reproducing the QM potential can be easily calculated as:

$$u(\varphi) = \frac{U(\varphi)}{N_Q} = \frac{U^{QM}(\varphi) - U^E(\varphi)}{N_Q} \quad (5)$$

As usual the simplest solution is never available: most MD/MM codes do not include a constrained energy minimization algorithm in their implementation, allowing for a direct calculation of $U^E(\varphi)$ and comparison with $U^{QM}(\varphi)$ (it is always good practice to parameterize the force field with the code that is used for running subsequent simulations): as a consequence, in the last years we always approximated the (internal) energy profile $U^E(\varphi)$ with the **free** energy profile $F^E(\varphi)$. Thermodynamics tells us that in the case of the NVT ensemble we normally use for such calculations, $F^E(\varphi) = U^E(\varphi) - TS^E(\varphi)$, where temperature and the entropy associated to molecular vibrational and translational modes appear. The approximation of $F^E(\varphi) \cong U^E(\varphi)$ [Johansson 2008] may be considered a good one for several reasons:

- we try to keep the temperature as low as possible (200 K)
- it is not the absolute value of $F^E(\varphi)$ or $U^E(\varphi)$ that matters, but the difference of energy between two different values of φ , and the derivative of the curves
- there are no strong physical arguments for having $S^E(\varphi)$ strongly depending on φ : for instance for biphenyl torsion the effect of considering entropy in ab initio calculations resulted in changes of the

minimum energy angle of about 5 degrees and in changes of energy of about 0.2 kJ/mol, both below the resolution of MM calculations [Johansson 2008]

- last but not least, all referees accepted this approximation so far ;)

The calculation of free energy is a long standing computational problem itself [Chipot 2007], however in our case, due to the relative simplicity of the torsional energy profile, at least two methods are available and affordable, although the latter is much quicker and more effective: Boltzmann inversion and Adaptive Biasing Force (ABF) calculations [Chipot 2007, Darve 2008, Henin 2010].

We start the truly tutorial part by applying the Boltzmann inversion method – the most “natural” one. In fact it consists in running a standard MD simulation at a given temperature, calculate from the trajectory the distribution of the dihedral angle of interest, and calculate the corresponding free energy profile by “inverting” the distribution, assuming that it is determined by Maxwell-Boltzmann’s statistics:

$$P(\varphi) \propto \exp[-F(\varphi)/kT] \quad (6)$$

$$F(\varphi) = -kT \log P(\varphi) + F_0 \quad (7)$$

where F_0 is set a posteriori in order to have only positive free energies with the minimum at $F = 0$.

Regarding the simulation setup, here we are interested in the comparison with QM calculations – i.e. isolated molecules – hence the best would be to simulate an isolated molecule with MD. However proper thermalization is difficult in these cases, and it is easy to risk the so-called “flying ice cube” problem [Note3]. We then prefer to run simulations in the gas phase, by adding a few Argon atoms which ensure a better exchange of kinetic energy by frequent collisions with the target molecule. Normally a box size of 50 with 40 Ar atoms is adequate (Figure 4). For T2, we use MP2/cc-pVTZ charges and “our” force for sexithiophene, largely coincident with the AMBER force field – see topology and parameter files in appendix b) and c).

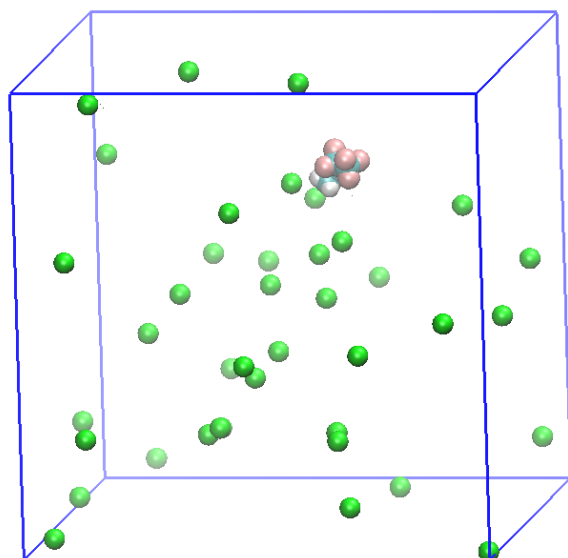


Figure 4: A typical simulation box with the target molecule and 40 Argon atoms in the gas phase.

The idea is to run a simulation long enough to be able to evaluate $U^E(\varphi)$ and then use equation 5) or a more elaborate one to derive the $u(\varphi)$ for each of the quadruples involved, here CA-CS-CS-CA, S-CS-CS-S, and 2 x CA-CS-CS-S. The proper length of the simulation depends on the temperature, the energy barrier, the level of accuracy needed, etc. We start with T=600 K and 10 ns; at this high T we expect that energy barriers can be passed rather easily. For this example, we opt for a dumb use of equation 5) and then we aim at finding a proper set of parameter for the X-CS-CS-X torsions: to estimate the corresponding $U^E(\varphi)$ the simplest way is to set their value to zero in the prm file for the first simulation “round”:

```
X CS CS X 0.0 2 180
```

We are then ready to start the simulation (see NAMD configuration file in appendix d). It is important to switch of

the Particle Mesh Ewald method because we are not interested in long range electrostatic interactions (all the charged atoms are inside the molecule) and because it considerably slows down the simulation. Once run it, the NAMD dcd file must have been converted as usual into the trj format (the outside ascii format, although a direct read from dcd is a planned improvement) via the command:

```
$ dcd_to_trj T2gas 1.
```

Then we use outside to compute the torsion angle distribution and free energy (see appendix e for an example input). Once the free energy $U^E(\varphi)$ is calculated, then equation 5 is used to compute the new force field parameters for the torsion. Normally the first correction is not sufficient to achieve convergence in the ff parameters, and then it is necessary to reiterate the procedure updating each time the ff parameters (summing the correction to the previous ones). Although other choices are possible, for performing fits and correction we typically use a gnuplot script (appendix f) which must be updated after each round and the parameter file as well. As already mentioned, in the script used in this tutorial equation 5 is adopted on purpose, even if is not completely correct, moreover a symmetrized form of the QM potential is also adopted ($U^{QM}(\varphi) = [2U^{QM}(\varphi) + 2U^{QM}(\varphi + 180)]/4$). In practice then at least two errors are contained in the script/procedure: we use the SS-CS-CS-SS distribution of the energy without taking into account that the other quadruples have not necessarily the same phase δ_i , while we do it for the QM potential, in practice decreasing its asymmetry (the peaks at 0 and 180 degrees have the same height in the symmetrized form). The wrong concept is that we try to fit an asymmetric dihedral potential with an implicitly symmetric function (the choice of a X-CS-CS-X expression in the parameter file): it could work only if N_Q would be equal to one. The results on the free energy profile of the first three iteration of this procedure are shown in figure 5: despite the conceptual error, things don't work too badly and already at the second round the FF produces a profile rather similar to the QM one. Unfortunately, further iterations (up to four were tried) do not bring any further improvement: this in our experience is the typical symptom of using the wrong symmetry (like here) or mathematical functions (for example, there is a typing error somewhere in the several files, or all δ_i are set to zero, or even n_i when odd values are requested, ...).

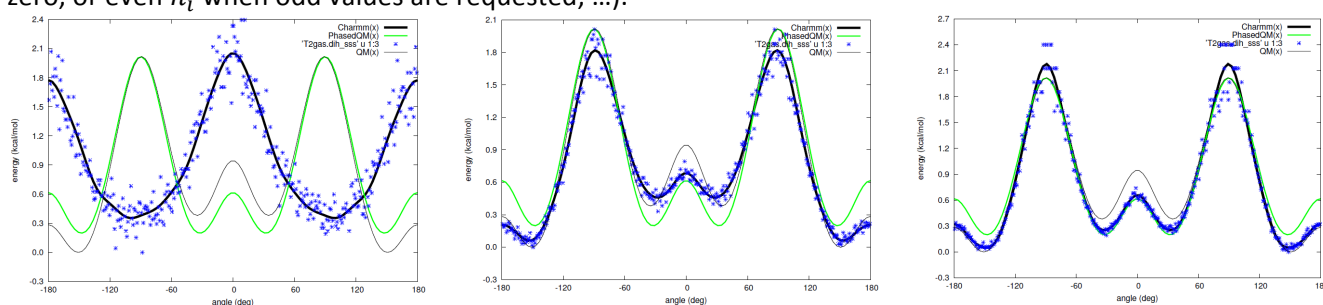


Figure 5: Thiophene-thiophene torsional free energy calculated from MD simulation (blue dots), corresponding fitting function (black line), and QM original and symmetrized “PhasedQM” potential (cyan and green lines). From the left, first “round” at 600 K, second at 300 K, and third at 200 K.

Figure 5 is still instructive as we can observe several features: i) what we called the “external” torsional potential $U^E(\varphi)$ is far from being negligible, and has peaks in correspondence with conformations on which the atoms A_i are eclipsed by B_i (figure 5 left [Note4]); ii) the noise in the simulation data depends on the temperature: a lower temperature gives lower noise in the populated, low energy regions, but also higher noise / worse sampling in the barrier regions; iii) fitting the simulated potential with an appropriate function somehow reduces the noise; iv) despite we use a FF expression symmetric in 0 and 180, the overall potential obtained from MD is not symmetric there: this because $U^E(\varphi)$ is not symmetric due to the different charges and van der Waals parameters for the SS and CS atom types.

For refining the FF, we turn to the second method for calculating $U^E(\varphi)$, i.e. to the adapting biasing force technique. To make it short, in this framework it is possible to define a collective variable (the torsion angle here) for which the free energy is calculated: i) the forces acting on φ are monitored and averaged during the MD simulation, and ii) equal and opposite forces are applied. Keeping track of the forces ($-\partial U(\varphi)/\partial \varphi$) for discrete values of φ allows for computing the free energy by integrating over φ (NAMD stores it in a file with extension .pmf), while the counterforces improve the sampling making easy to overcome the barriers and

determining an almost uniform sampling at the end of the simulation. ABF calculations are available in NAMD form version 2.7 onwards, to run our one we just need to add to the configuration file the two following lines:

```
colvars          on
colvarsConfig    SStors.colvars
```

We also need to provide an additional file "SStors.colvars" where the collective variable is defined – see appendix g for our torsion and NAMD manual for more elaborated cases. Regarding the MD simulation, it is completely similar to the previous ones, albeit shorter simulation times (2 ns) are normally adequate, and although we are no more worried on sampling, we cannot use very low temperatures as Ar atoms could crystallize – we continue with 200 K. As now we intend to consider that the QM potential has different values at 0 and 180 degrees, we choose of modifying in the parameter file only the SS CS CS SS parameters while leaving the other unaltered with respect to figure 5 right, where we had (a decent guess):

```
X  CS CS X  -0.1229 1 0.0
X  CS CS X  -0.352  2 0.0
X  CS CS X   0.1015 4 0.0
X  CS CS X   0.0141 6 0.0
```

And now we distinguish some quadruples and switch to:

```
CA CS CS X  -0.1229 1 0.0
CA CS CS X  -0.352  2 0.0
CA CS CS X   0.1015 4 0.0
CA CS CS X   0.0141 6 0.0
SS CS CS SS  0.0    1 0.0
```

and run the first ABF/MD simulation. To analyze the results, we must modify the gnuplot script by changing the file name and the columns of the simulation data (the .pmf file, column 1:2), and most importantly change the dihedral multiplicity from 4 to 1 as now we are dealing with the SS CS CS SS quadruple only, and fit the original QM function instead of a symmetrized one. After this run, the force field potential is still far away from the QM one (see figure 6, left), as the SS CS CS SS were set to zero. By fitting again the difference we obtain the following parameters to start the second ABF/MD run:

```
SS  CS CS SS  -0.4363 2 0.0
SS  CS CS SS   0.0603 3 0.0
SS  CS CS SS   0.1334 4 0.0
SS  CS CS SS  -0.0159 8 0.0
```

The resulting potential is shown in figure 6 right: finally the agreement between QM and MM is satisfactory and no further runs are necessary.

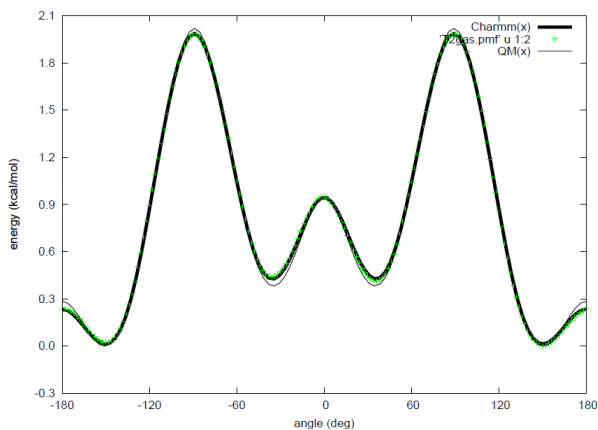
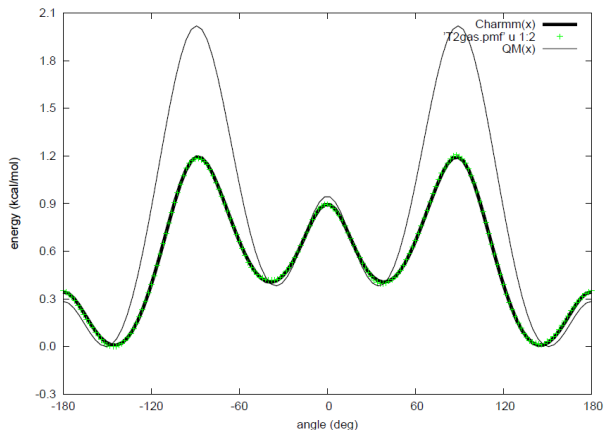


Figure 6: Thiophene-thiophene torsional free energy calculated with ABF/MD simulation (green dots), corresponding fitting function (thick line), and QM potential (thin line). Left first and right second “round” at 200 K.

We conclude this tutorial with a quick comparison between the Boltzmann inversion and ABF techniques applied to the dithiophene torsion, showing a few distributions and energy profiles in figure 7 and discussing them:

- unbiased MD gives the real distribution while ABF a flat one with almost uniform sampling
- MD is limited by a poor sampling in the high energy regions, and to overcome this effect rather long runs (50 ns) are necessary
- the free energy profiles calculated with MD are very similar at 300 and 600 K, indicating that the entropy dependence on the dihedral angle is negligible; besides, the noise is much higher at 600 K
- ABF gives a very smooth and well defined free energy profile.

In summary, the ABF method is much more efficient when applied to the current problem and must be preferred when available, but the effectiveness of both methods relies in a proper definition of the torsion angle symmetry.

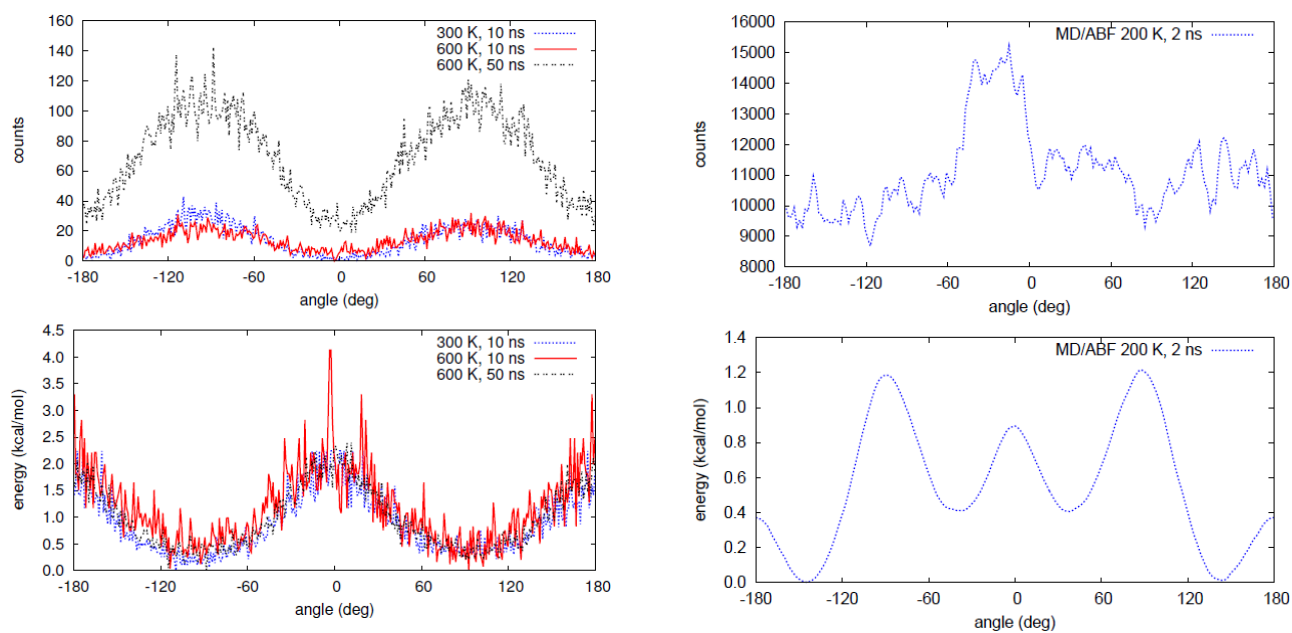


Figure 6: Comparison between MD/BI (left) and ABF/MD (right) distributions (top) and free energy profiles (bottom) for the simulated system shown in figure 3.

References

- [Becke 1993] A. D. Becke, **"Density-functional thermochemistry. III. The role of exact exchange"**, *J. Chem. Phys.* 98, 5648-52 (1993)
- [Berardi 2005] R. Berardi, G. Cainelli, P. Galletti, D. Giacomini, A. Gualandi, L. Muccioli, C. Zannoni, **"Can the pi-facial selectivity of solvation be predicted by atomistic simulation?"**, *J. Am. Chem. Soc.* 127, 10699-10706 (2005)
- [Chai 2008] J.-D. Chai and M. Head-Gordon, **"Long-range corrected hybrid density functionals with damped atom-atom dispersion corrections"**, *Phys. Chem. Chem. Phys.* 10, 6615 (2008)
- [Chipot 2007] C. Chipot, A. Pohorille editors, **"Free energy calculations. Theory and Applications in Chemistry and Biology"**, Springer (2007) – available in the small LCgroup library
- [Darve 2008] E. Darve, D. Rodriguez-Gomez, A. Pohorille, **"Adaptive biasing force method for scalar and vector free energy calculations"**, *J. Chem. Phys.* 128, 144120 (2008)
- [Grimme 2006] S. Grimme, **"Semiempirical GGA-type density functional constructed with a long-range dispersion correction"**, *J. Comp. Chem.* 27, 1787-99 (2006)
- [Henin 2010] J. Hénin, J. Fiorin, C. Chipot, M. L. Klein **"Exploring Multidimensional Free Energy Landscapes Using Time-Dependent Biases on Collective Variables"**, *J. Chem. Theory Comput.* 6, 35-47 (2010)
- [Johansson 2008] M. I. P. Johansson, J. Olsen, **"Torsional Barriers and Equilibrium Angle of Biphenyl: Reconciling Theory with Experiment"**, *J. Chem. Theory Comput.* 4, 1460–1471 (2008)
- [Jorgensen 1996] W. L. Jorgensen, D. S. Maxwell, J. Tirado-Rives, J. **"Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids"**, *J. Am. Chem. Soc.* 118, 11225-11236 (1996)
- [Note1] see <http://en.wikipedia.org/wiki/Pseudoscalar>
- [Note2] see <http://www.charmm.org/documentation/c36b1/parmfile.html>
- [Note3] see http://en.wikipedia.org/wiki/Flying_ice_cube Harvey SC, Tan RKZ, Cheatham TE. (1998). The flying ice cube: Velocity rescaling in molecular dynamics leads to violation of energy equipartition. *J Comp Chem* 19(7): 726-40. See here for complete derivation and examples.
- [Note4] see http://en.wikipedia.org/wiki/Eclipsed_conformation
- [Pizzirusso 2011] A. Pizzirusso, M. Savini, L. Muccioli, C. Zannoni **"An atomistic simulation of the liquid-crystalline phases of sexithiophene"**, *J. Mater. Chem.* 21, 125-133 (2011)
- [Riley 2012] K. E. Riley, J. A. Platts, J. Řezáč, P. Hobza, J. Grant Hill **"Assessment of the Performance of MP2 and MP2 Variants for the Treatment of Noncovalent Interactions"**, *J. Phys. Chem. A* 116, 4159–4169 (2012)
- [Roscioni 2012] O. M. Roscioni, **"Electronic processes at silicon-based interfaces"** *Minotor project Deliverable D3.5* (2012)
- [Sancho-García] J. C. Sancho-García, A. J. Pérez-Jiménez, **"Assessment of double-hybrid energy functionals for pi-conjugated systems"**, *J. Chem. Phys.* 131, 084108 (2009)
- [Schwabe 2007] T. Schwabe and S. Grimme, **"Double-hybrid density functionals with long-range dispersion corrections: higher accuracy and extended applicability"**, *Phys. Chem. Chem. Phys.* 9, 3397 (2007)
- [Wang 2004] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, D. A. Case **"Development and Testing of a General Amber Force Field"**, *J. Comput. Chem.* 24, 1157-1173 (2004)

Appendix

a) Fortran code for the calculation of a dihedral angle

```
function DiedroV (v12,v23,v34) result (D)
  ! calculates a dihedral angle in degrees
  ! cos(1234)= (r12 x r23) (r23 x r34) / module
  real(rk),intent(in):: v12(3),v23(3),v34(3)
  real(rk):: D
  real(rk):: p123(3),p234(3),sign_check(3)
  real(rk):: coseno,segno,n123,n234

  p123(1)=v12(2)*v23(3)-v12(3)*v23(2)
  p123(2)=v12(3)*v23(1)-v12(1)*v23(3)
  p123(3)=v12(1)*v23(2)-v12(2)*v23(1)
  p234(1)=v23(2)*v34(3)-v23(3)*v34(2)
  p234(2)=v23(3)*v34(1)-v23(1)*v34(3)
  p234(3)=v23(1)*v34(2)-v23(2)*v34(1)

  n123=sqrt(dot_product(p123,p123))
  n234=sqrt(dot_product(p234,p234))
  p123=p123/n123
  p234=p234/n234

  coseno=dot_product(p123,p234)

  sign_check(1)=p123(2)*p234(3)-p123(3)*p234(2)
  sign_check(2)=p123(3)*p234(1)-p123(1)*p234(3)
  sign_check(3)=p123(1)*p234(2)-p123(2)*p234(1)

  segno=dot_product(sign_check,v23)
  segno=segno/abs(segno)

  D=segno*rad_to_deg*acos(coseno)
end function DiedroV
```

b) CHARMM topology file for dithiophene and argon

```
* Charmm tpg file from uniteda
AUTOgenerate ANGLES DIHEDRAL
MASS 1 CA 12.0110 CA
MASS 2 CS 12.0110 CS
MASS 3 HA 1.0079 HA
MASS 4 SS 32.0655 SS
MASS 5 AR 40.0000 AR

RESIDUE T2 0.0
*
*
*          H16      H15
*          /      \
*          S7      C1 - C5
*          /      \
* H13 C8      C6 - C2      C4 H14
*          \      /      \
*          C9 - C10      S3
*          H12      H11
group
atom C1 CA -0.08645
atom C2 CS -0.01165
atom S3 SS 0.0407
atom C4 CS -0.24045
atom C5 CA -0.17435
atom H14 HA 0.2051
```

```

atom H15    HA    0.1527
atom H16    HA    0.11445
group
atom C10    CA   -0.08645
atom C6     CS   -0.01165
atom S7     SS    0.0407
atom C8     CS   -0.24045
atom C9     CA   -0.17435
atom H13    HA    0.2051
atom H12    HA    0.1527
atom H11    HA    0.11445

bond C1     C2     C1     C5     C1     H16     C2     S3
bond C2     C6     S3     C4     C4     C5     C4     H14
bond C5     H15    C6     S7     C6     C10    S7     C8
bond C8     C9     C8     H13    C9     C10    C9     H12     C10     H11

RESIDUE AR   0.0
group
atom AR     AR   .00000
end

```

c) CHARMM parameter file for dithiophene and argon

```

BOND
HA CA      344.3   1.08
HA CS      344.3   1.077
CA CA      478.4   1.42
CA CS      478.4   1.38
SS CS      256.6   1.83
CS CS      478.4   1.43
ANGLE
HA CA CA   35.0 123.
HA CA CS   35.0 123.
HA CS CA   35.0 128.
HA CS SS   44.3 120.
CA CA CS   63.0 120.
CA CS CS   63.0 120.
CA CS SS   62.0 111.
CS CS SS   62.0 120.
CS SS CS   63.3  92.
DIHEDRAL
x  CA CA x  3.6250 2 180.
x  CA CS x  3.6250 2 180.
x  CS SS x  1.2000 2 180.
x  C2 CS x  0.0000 2 180.
x  CS CS x  0.0 2 180.
NONBONDED
HA 0. -0.015 1.4590 0. -0.0075 1.4590
CA 0. -0.086 1.9080 0. -0.043 1.9080
CS 0. -0.086 1.9080 0. -0.043 1.9080
SS 0. -0.250 2.0000 0. -0.125 2.0000
AR 0. -.2300 1.911 0. .0 0.

```

d) NAMD configuration file for dithiophene and argon

```

#--- integrator
numsteps          50000000
timestep          1
nonbondedFreq    1
fullElectFrequency 1
stepspercycle    20

```

```

#--- Parameter options (14 scaling is for electrostatics)
paratypeCharmm on
parameters T2.prm
structure T2gas.psf
exclude scaled1-4
1-4scaling 0.8333333
cutoff 12.
switching on
switchdist 10.
pairlistdist 15.0

#--- Thermodynamic
rescaleTemp 300K
rescalefreq 100
COMmotion no
BerendsenPressure off

#--- PBC
cellBasisVector1 50. 0. 0.
cellBasisVector2 0. 50. 0.
cellBasisVector3 0. 0. 50.

#--- PME
dielectric 1
PME off

#--- Input coords
temperature 300K
coordinates T2gas.pdb
#bincoordinates T2gasr.coor
#binvelocities T2gasr.vel

#--- Output & Restart

binaryoutput no
outputname T2gas

binaryrestart yes
restartname T2gasr
restartfreq 100000

DCDfile T2gas.dcd
DCDfreq 2000
XSTfreq 2000

#--- Standard Output
outputEnergies 10000

```

e) Outside input for computing the dihedral angle distribution and energy for the X CS CS X torsion force

```

--- reading the template pdb file
100
T2gas.pdb
--- reading topology and parameters
998
T2.tpc
T2.prm
0.5
0.833333
--- setting temperature
98
200.
--- reading trajectory
99

```

```

T2gas.trj
--- calculating torsion angle distribution and energy
6
T2
S3
C2
C6
S7
sss
--- exiting
0

```

f) Gnuplot script for fitting the X CS CS X torsional force field

```

set ang deg # very important
set xtics 60
set ytics 0.3
set format y "%1.1f"

n=4 # multiplicity of the dihedral: very important!

set xlabel "angle (deg)"
set ylabel "energy (kcal/mol)"

# QM(x) is the function fitting the QM calculated torsional profile
QM(x)= o0+o1*(1+cos(x+na1))+ o2*(1+cos(2*x+na2)) + o3*(1+cos(3*x+na3)) + \
o4*(1+cos(4*x+na4)) +o5*(1+cos(5*x+na5)) + o6*(1+cos(6*x+na6)) + \
o7*(1+cos(7*x+na7)) + o8*(1+cos(8*x+na8))+ o9*(1+cos(9*x+na9)) + \
o10*(1+cos(10*x+na10)) + o11*(1+cos(11*x+na11))+o12*(1+cos(12*x+na12)) + \
o13*(1+cos(13*x+na13)) +o14*(1+cos(14*x+na14))
na1 = 0. ; na2 = 0. ; na3 = 0. ; na4 = 0. ; na5 = 0. ; na6 = 0. ; na7 = 0. ;
na8 = 0. ; na9 = 0. ; na10 =0. ; na11 =0. ; na12 =0. ; na13 =0. ; na14 =0.

fit QM(x) './_mp2ccpvtz/ene.dat' u 1:3 via
o0,o1,o2,o3,o4,o5,o6,o7,o8,o9,o10,o11,o12,o13,o14
#,na1,na2,na3,na4,na5,na6,na7,na8,na9,na10,na11,na12,na13,na14

# PhasedQM(x) is what we expect to obtain by taking into account the symmetry and the phases
# of the four quadruples involved: here it is used replacing QM(x) to show that nevertheless
# we do not obtain a symmetric Charmm term

PhasedQM(x)= ( 2*QM(x) + 2*QM(x+180))/n

pl [-180:180] QM(x), './_mp2ccpvtz/ene.dat' u 1:3,PhasedQM(x)
pause -1

# Charmm(x) is the function fitting the simulated torsional profile contained in
'T2gas.dih_sss'
Charmm(x)= c0+(c1)*(cos(x+a1))+(c2)*(cos(2*x+a2))+(c3)*(cos(3*x+a3))+(c4)*(cos(4*x+a4)) + \
(c5)*(cos(5*x+a5))+(c6)*(cos(6*x+a6))+(c7)*(cos(7*x+a7))+(c8)*(cos(8*x+a8)) + \
(c9)*(cos(9*x+a9))+(c10)*(cos(10*x+a10))+(c11)*(cos(11*x+a11))+(c12)*(cos(12*x+a12))
c0 =0.1 ; c1 =0.1 ; c2 =0.1 ; c3 =0.1 ; c4 =0.1 ; c5 =0.1 ; c6 =0.1 ; c7 =0.1
c8 =0.1 ; c9 =0.1 ; c10=0.1 ; c11=0.1 ; c12=0.1
a1 =0.000001 ; a2 =0.000001 ; a3 =0.000001 ; a4 =0.000001 ; a5 =0.000001
a6 =0.000001 ; a7 =0.000001 ; a8 =0.000001 ; a9 =0.000001 ; a10=0.000001
a11=0.000001 ; a12=0.000001

fit Charmm(x) 'T2gas.dih_sss' u 1:3 via c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12

# the correct way should be to take into account of the dihedral geometry also here,
# by imposing Charmm_total(x)= ( 2*Charmm(x) + 2*Charmm(x+180))/n and use Charmm_total
# in the following

pl [-180:180][*:*] \

```

```

Charmm(x) lw 8 lt 7, PhasedQM(x) lw 4, 'T2gas.dih_sss' u 1:3, QM(x) lw 2 lt 7
pause -1

# odiff(x) at this point contains the force field used in the MD simulations
# at the first "round" it is zero, then it must be updated after each change in the prm file
odiff(x)=odi0+(odi1)*cos(1*x+oai1)+(odi2)*cos(2*x+oai2)+(odi3)*cos(3*x+oai3)+\
(odi4)*cos(4*x+oai4)+(odi5)*cos(5*x+oai5)+(odi6)*cos(6*x+oai6)+(odi7)*cos(7*x+oai7)+\
(odi8)*cos(8*x+oai8)+(odi9)*cos(9*x+oai9)+(odi10)*cos(10*x+oai10)+\
(odi11)*cos(11*x+oai11)+(odi12)*cos(12*x+oai12)

odi0 =0.0 ; odi1 =0.0 ; odi2 =0.0 ; odi3 =0.0 ; odi4 =0.0 ; odi5 =0.0 ; odi6 =0.0
odi7 =0.0 ; odi8 =0.0 ; odi9 =0.0 ; odi10=0.0 ; odi11=0.0 ; odi12=0.0
oai0 =0.0 ; oai1 =0.0 ; oai2 =0.0 ; oai3 =0.0 ; oai4 =0.0 ; oai5 =0.0 ; oai6 =0.0
oai7 =0.0 ; oai8 =0.0 ; oai9 =0.0 ; oai10=0.0 ; oai11=0.0 ; oai12=0.0

# example of updated values of odiff(x) after some "rounds": only the non zero values
# need to be specified
odi0 =-0.1474
odi1 =-0.1281
odi2 =-1.4604
odi4 =0.306
odi8 =0.019
odi11=0.018
odi12=-0.0161

# we fit the difference between the symmetric PhasedQM(x) instead of QM(x)- if this was not
# a tutorial, we should fit QM(x) or PhasedQM(x)? I am not sure of the answer ;) Let's say
# that the geometry of the dihedral must be taken into account for both, QM(x) and Charmm(x)
set table "t1"
pl [-180:180] PhasedQM(x)-Charmm(x)
unset table

# diff(x) contains in a first instance the difference from current force field results
# and QM(x), (first fit command). At last, it contains this difference plus the current
# force field used in the prm file (second fit command)
diff(x)=(di0+ (di1)*cos(1*x+ai1) + (di2)*cos(2*x+ai2) + (di3)*cos(3*x+ai3) +
(di4)*cos(4*x+ai4) + (di5)*cos(5*x+ai5)\
+ (di6)*cos(6*x+ai6) + (di7)*cos(7*x+ai7) + (di8)*cos(8*x+ai8) +
(di9)*cos(9*x+ai9) + (di10)*cos(10*x+ai10) + (di11)*cos(11*x+ai11)+ (di12)*cos(12*x+ai12) )

di0 =0.1 ; di1 =0.1 ; di2 =0.1 ; di3 =0.1 ; di4 =0.1 ; di5 =0.1 ; di6 =0.1
di7 =0.1 ; di8 =0.1 ; di9 =0.1 ; di10=0.1 ; di11=0.1 ; di12=0.1

ai0 =0.00001 ; ai0 =0.00001 ; ai1 =0.00001 ; ai2 =0.00001 ; ai3 =0.00001 ; ai4 =0.00001 ;
ai5 =0.00001 ;
ai6 =0.00001 ; ai7 =0.00001 ; ai8 =0.00001 ; ai9 =0.00001 ; ai10=0.00001 ; ai11=0.00001 ;
ai12=0.00001

fit diff(x) 't1' u 1:($2) via di0,di1,di2,di3,di4,di5,di6,di7,di8,di9,di10,di11,di12
#fit diff(x) 't1' u 1:($2) via di0,di1,di2,di3,di4,di5,di6,di7,di8,di9,di10,di11,di12,\
#
ai1,ai2,ai3,ai4,ai5,ai6,ai7,ai8,ai9,ai10,ai11,ai12

pl [-180:180] diff(x),'t1'
pause -1

set table "t2"
pl [-180:180] diff(x)+odiff(x)
unset table

fit diff(x) 't2' u 1:($2) via di0,di1,di2,di3,di4,di5,di6,di7,di8,di9,di10,di11,di12
#fit diff(x) 't2' u 1:($2) via di0,di1,di2,di3,di4,di5,di6,di7,di8,di9,di10,di11,di12,\
#
ai1,ai2,ai3,ai4,ai5,ai6,ai7,ai8,ai9,ai10,ai11,ai12

```

```

pl [-180:180] PhasedQM(x),QM(x), Charmm(x) t "sim",Charmm(x)+diff(x)-odiff(x)+0.1 t
"corrected",diff(x) t "force field"
pause -1

# here the values of diff(x) are conveniently plotted and formatted so to be easily
# copy pasted into the prm file and into the gnu file for the following "round"

fmt(x)= int(10000*x+.00001)/10000.

sign(x) = abs(x)/x
nint(x) = int( sign(x)*0.5+x)
angle(x)= x -360*nint(x/360)

print "Reminder: please select only up to 6 dihedrals!!! "

# this is a minimum value to exclude negligible cosine terms
# the lines printed here must be copied in the prm file to replace previous ones
minV=0.05/n

if( abs(fmt(di1/n) )> minV) { print "X CS CS X ",fmt(di1/n),1,fmt(angle(-ai1)) }
if( abs(fmt(di2/n) )> minV) { print "X CS CS X ",fmt(di2/n),2,fmt(angle(-ai2)) }
if( abs(fmt(di3/n) )> minV) { print "X CS CS X ",fmt(di3/n),3,fmt(angle(-ai3)) }
if( abs(fmt(di4/n) )> minV) { print "X CS CS X ",fmt(di4/n),4,fmt(angle(-ai4)) }
if( abs(fmt(di5/n) )> minV) { print "X CS CS X ",fmt(di5/n),5,fmt(angle(-ai5)) }
if( abs(fmt(di6/n) )> minV) { print "X CS CS X ",fmt(di6/n),6,fmt(angle(-ai6)) }
if( abs(fmt(di7/n) )> minV) { print "X CS CS X ",fmt(di7/n),7,fmt(angle(-ai7)) }
if( abs(fmt(di8/n) )> minV) { print "X CS CS X ",fmt(di8/n),8,fmt(angle(-ai8)) }
if( abs(fmt(di9/n) )> minV) { print "X CS CS X ",fmt(di9/n),9,fmt(angle(-ai9)) }
if( abs(fmt(di10/n) )> minV) { print "X CS CS X ",fmt(di10/n),10,fmt(angle(-ai10)) }
if( abs(fmt(di11/n) )> minV) { print "X CS CS X ",fmt(di11/n),11,fmt(angle(-ai11)) }
if( abs(fmt(di12/n) )> minV) { print "X CS CS X ",fmt(di12/n),12,fmt(angle(-ai12)) }

# what is printed here must instead be copied after the odiff(x) definition above if we
# want to refine the fit in with a second round of simulations

print "odi0 =",fmt(di0 )
if( abs(fmt(di1/n) )> minV) {print "odi1 =",fmt(di1 ) }
if( abs(fmt(di2/n) )> minV) {print "odi2 =",fmt(di2 ) }
if( abs(fmt(di3/n) )> minV) {print "odi3 =",fmt(di3 ) }
if( abs(fmt(di4/n) )> minV) {print "odi4 =",fmt(di4 ) }
if( abs(fmt(di5/n) )> minV) {print "odi5 =",fmt(di5 ) }
if( abs(fmt(di6/n) )> minV) {print "odi6 =",fmt(di6 ) }
if( abs(fmt(di7/n) )> minV) {print "odi7 =",fmt(di7 ) }
if( abs(fmt(di8/n) )> minV) {print "odi8 =",fmt(di8 ) }
if( abs(fmt(di9/n) )> minV) {print "odi9 =",fmt(di9 ) }
if( abs(fmt(di10/n) )> minV) {print "odi10=",fmt(di10) }
if( abs(fmt(di11/n) )> minV) {print "odi11=",fmt(di11) }
if( abs(fmt(di12/n) )> minV) {print "odi12=",fmt(di12) }

if( abs(fmt(di1/n) )> minV) {print "oai1 = ",fmt(angle(ai1)) }
if( abs(fmt(di2/n) )> minV) {print "oai2 = ",fmt(angle(ai2)) }
if( abs(fmt(di3/n) )> minV) {print "oai3 = ",fmt(angle(ai3)) }
if( abs(fmt(di4/n) )> minV) {print "oai4 = ",fmt(angle(ai4)) }
if( abs(fmt(di5/n) )> minV) {print "oai5 = ",fmt(angle(ai5)) }
if( abs(fmt(di6/n) )> minV) {print "oai6 = ",fmt(angle(ai6)) }
if( abs(fmt(di7/n) )> minV) {print "oai7 = ",fmt(angle(ai7)) }
if( abs(fmt(di8/n) )> minV) {print "oai8 = ",fmt(angle(ai8)) }
if( abs(fmt(di9/n) )> minV) {print "oai9 = ",fmt(angle(ai9)) }
if( abs(fmt(di10/n) )> minV) {print "oai10= ",fmt(angle(ai10)) }
if( abs(fmt(di11/n) )> minV) {print "oai11= ",fmt(angle(ai11)) }
if( abs(fmt(di12/n) )> minV) {print "oai12= ",fmt(angle(ai12)) }

# this is an example of what is printed in this last part
Reminder: please select only up to 6 dihedrals!!!

```

```
X CS CS X -0.0332 1 0.0
X CS CS X -0.3637 2 0.0
X CS CS X 0.0769 4 0.0
X CS CS X -0.003 6 0.0
X CS CS X 0.0062 8 0.0
odi0 =-0.1532
odi1 =-0.1329
odi2 =-1.4549
odi4 =0.3077
odi6 =-0.0121
odi8 =0.0248
oai1 = 0.0
oai2 = 0.0
oai4 = 0.0
oai6 = 0.0
oai8 = 0.0
```

g) NAMD collective variable configuration file for thiophene-thiophene torsion angle

```
colvar {
  name phi

  width 2.

  lowerBoundary -180.
  upperBoundary +180.

  # S3 C2 C6 S7
  dihedral {
    group1 { atomNumbers 3}
    group2 { atomNumbers 2}
    group3 { atomNumbers 10}
    group4 { atomNumbers 11}
  }
}

abf {
  colvars phi
  fullSamples 100
}
```