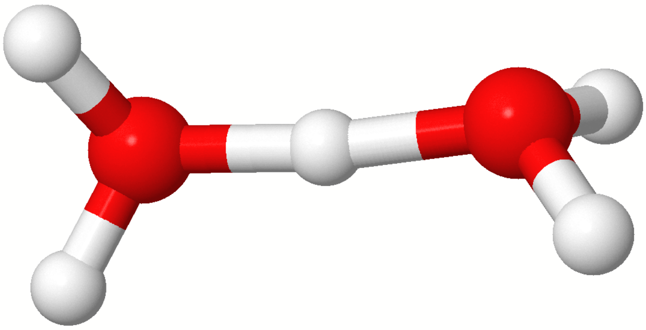
Special Course in Theoretical Physics
PHYS-E0546: DFT for Experts

Aalto University, September 2019



Tutorial 4: DFT with Molecules
Molecular Dynamics and Dispersion Corrections

Manuscript for Exercise Problems

Prepared by Ondřej Krejčí, Orlando Silveira and Adolfo Fumega
Developed by Patrick Rinke, Milica Todorović and Xi Chen
Department of Applied Physics
Aalto University, 2022

A quick summary of the exercises

In this tutorial we address DFT methodology typically employed with molecular systems. Part I introduces the basic concepts of molecular dynamics and allows you to run and analyse a fully quantum mechanical MD calculation. Parts II and III tackles dispersion corrections. In previous tutorials we employed gradient-corrected or hybrid exchange–correlation functionals. These might work very well for covalent, ionic and metallic bonds and to some degree even for intermolecular interactions, but they certainly miss dispersion (van der Waals) interactions. These are often relatively weak but always attractive, purely quantum forces which exist between every two (or more) fragments of matter and thus become important for large and realistic systems.

For every exercise, we also provide solutions and sample input files. They can be found in `/work/courses/unix/PHYS/E0546/TUTORIALS/Tutorial_4` directory. In order to study realistic systems, the example calculations today are computationally intensive. We strongly recommend to use the python scripts provided to minimise input preparation and data analysis time. Do study the script structure, input and output files. At the end of each exercise, please attempt to interpret the results from the point of view of the level of theory employed in the calculations.

Additional tools and programs

- **Visualization tools:**
  To visualize structures, vibrational modes, charge density plots, etc., several programs including molden, vmd, and jmol are installed on your workstations. For molecular dynamics trajectories, you are encouraged to use VMD or jmol.

- **Utility scripts:**
  DFT output files are complex, so we use utility scripts to extract important data. In the `/work/courses/unix/PHYS/E0546/CODE/utilities` directory you will find tools like `aims-SCF_convergence.awk` and `get_relaxation_info.pl` to monitor the simulation, as well as `create_xyz_movie.pl` to track the geometry optimisation. In this tutorial, additional scripts can be found in:
  `/work/courses/unix/PHYS/E0546/TUTORIALS/Tutorial_4/utilities`.

Part I: Molecular dynamics calculations

With this tutorial, we provide a brief introduction to state-of-the-art techniques for sampling the potential energy surface of a system at constant energy (microcanonical ensemble). The technique we adopt is molecular dynamics (MD), which can be considered as old as modern physics itself, since we *propagate* in time a system of particles, given initial positions and momenta, by *numerically integrating* Newton's equations of motion for the system.

The original formulation of Newton's second law is [1]: "*Lex II: Mutationem motus proportionalem esse vi motrici impressae, et fieri secundum lineam rectam qua vis illa imprimitur.*", in modern English, "Law II: The change of momentum of a body is proportional to the impulse impressed on the body, and happens along the straight line on which that impulse is impressed." In symbols:

$$\int_{\Delta t} \mathrm{F} dt = \Delta \mathrm{p} \tag{1}$$

Where the vector **F** is the force acting on a body, **p** is its momentum, Δt is the time along which the force is applied. The kind of MD we are interested in is Born-Oppenheimer MD, where nuclei are treated as classical particles and the forces between them are evaluated by solving the (ground-state) electronic structure problem for a given number of electrons and the nuclei at given positions.

Educational Objectives

- Perform a simple DFT molecular dynamics calculation.
- Analyse the microcanonical ensemble stability and trajectory.

Problem I: The microcanonical ensemble: setting the time step

To start our first Molecular Dynamics (MD) simulation, we will use a simple system, $H_5O_2^+$ which some of you may recognize as the Zundel (cat)ion. This system is one of the (protonated) water clusters that are thought to be the "building blocks" of liquid water. [2] The exercise below poses the problem of choosing the right settings for obtaining a stable and reliable molecular dynamics trajectory.

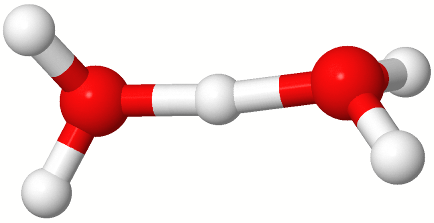


Figure 1: The Zundel cation

We will investigate the effects of the time step size for the integration of the equations of motion in a microcanonical molecular dynamics simulation. For a better illustration, we will not only consider the $H_5O_2^+$ cluster, but also its heavier, deuterated counterpart $D_5O_2^+$ [*]. Geometry files, containing structures and velocities for both $H_5O_2^+$ and $D_5O_2^+$ are already provided in the folder `/work/courses/unix/PHYS/E0546/TUTORIALS`/Tutorial_4/.
In order to substitute hydrogen with deuterium, you should substitute hydrogen's by deuterium's `species` and `mass` in the `control.in` like this:

```
species D
mass 2.014102
```

In general the `species` label should be changed such that the same label appears in the `geometry.in`

[*] The deuterium atom has one proton and one neutron in its nucleus, thus about twice as heavy as hydrogen.

file. Note that this file also contains velocities assigned to each atom, so that the molecular dynamics run will not use a random initialization. These velocities come from a previous equilibration of the molecule at ∼300K.

Tasks

1. Copy the geometry file corresponding to the molecule you will calculate to `geometry.in`. You may with to start with $H_5O_2^+$ and continue with $D_5O_2^+$.

2. Now we generate the basic `control.in` file for MD. Use the LDA (**pw-lda**) functional, no spin polarization (**spin none**) or relativity and Gaussian smearing of 0.01. Remember that the Zundel cation carries an additional proton charge. For the H and O elements, choose light settings with a tier 1 basis set.

3. Next, set the following "accurate" self-consistency convergence criteria:
```
mixer pulay
sc_iter_limit 100
sc_accuracy_rho 1E-5
sc_accuracy_eev 1E-4
sc_accuracy_etot 1E-6
sc_accuracy_forces 5E-4
```

4. Lastly, specify the MD control flags like this:
```
MD_run TOTAL_TIME NVE
MD_time_step TIME_STEP
wf_extrapolation none
MD_clean_rotations .false.
MD_restart .false.
```

5. Choose a 0.15 ps MD run in the microcanonical ensemble, using a 0.0005 ps ($\Delta t = 0.5$ fs) time step (flags `MD_run` and `MD_time_step` respectively). All time variables are expected in units of ps.

6. Start a parallel calculation on 6-8 CPUs. It should take less than 5min. You can check the progress by typing: `grep "Simulation time" "Output-File" | wc -l`

   TIP:
   You can find the amount of CPU cores by running a command: `lscpu` .
   Check the 4th line of the output.

7. When it is done, use the "process_MD.py" script to analyze your run by typing:
```
python3 process_MD.py "Output-File"
```
   The script will ask which ensemble you used in the simulations. Type "NVE" (microcanonical ensemble) and then the script will generate an output file (*.MD.out) in the current directory.

8. Plot the total energy (fifth column of the script output file) vs. the simulation time (first column of the script output file) with `xmgrace`, `gnuplot` or prepared python script for plotting with matploltlib. If you decide for the the last (by far, easiest solution), please type:
```
python3 plot_MD.py "column#" file1 file2 file3
```
   It will generate MD_run.png file in the current directory. With this script you can plot multiple files simultaneously or plot other quantities from the .dat file. Please note that column numbering starts from 1. Please feel free to modify the script to your taste. How would you describe what you see? Was this somewhat predictable?

9. Increase the time step (Δt) of the MD simulation to 0.001 ps. Run FHI-aims (total of 0.15 ps) and keep the output.

10. Increase the time step to 0.002 ps and run the simulation again, redirecting the output to another file.

11. Together with the plot for $\Delta t = 0.0005\,\mathrm{ps}$, plot the total energy vs. simulation time using `plot_MD.py` script for $\Delta t = 0.001\,\mathrm{ps}$ and $\Delta t = 0.002\,\mathrm{ps}$.

    How do the energy fluctuations develop? Do you notice something strange happening for the $\Delta t = 0.002\,\mathrm{ps}$ run?

12. For the last MD run, inspect the dynamics of the system by running
    `create_xyz_movie.pl "FHI-aims-output-file" > "script-output-file".xyz`
    and visualise the movie with VMD of Jmol. What can you observe about the system?

13. Now repeat the calculation with the alternative system ($D_5O_2^+$) following the same methodology. Plot the energies for the different timesteps and visualise the trajectories. Are there differences in the behaviour of the two systems?

From the practical point of view, a larger time step is desirable, since it allows to assess longer trajectories in shorter computational times. Notice, however, that the $\Delta t = 0.002\,\mathrm{ps}$ simulation diverges for $H_5O_2^+$. In fact, the molecule dissociates. The reason for the dissociation is that the integrator is unable to deal with these "big" time steps. This integrator uses a simple Verlet algorithm [3], where the error in the trajectory goes with Δt^4.

If you are simulating $D_5O_2^+$, the $\Delta t = 0.002\,\mathrm{ps}$ simulation might stay intact, although the oscillations are unacceptable for proper production run. The fact that it does not explode illustrates an important point: the largest Δt that can be used in a particular integration algorithm depends on the highest vibrational frequency of the system. Since the D atoms, being heavier, have a larger vibrational period (do you understand why this is obvious?), the used Δt can also be larger.

Important MD advice:
Just like we converge settings in T=0K ground state DFT calculations to obtain accurate results (see bulk Si example), we take care to achieve 'convergence' in the macroscopic quantities that characterise thermalised MD calculations. These can be the energy (E), volume (V), temperature (T) and pressure (P), depending on the type of MD calculation (ensemble). In the example of microcanonical ensemble (NVE) above, number of atoms N, V and E stay constant during the calculation. The cannonical ensemble is (NVT), and in the grand cannonical (NPT) ensemble temperature and pressure should be constant. It is important to monitor these variables throughout the MD calculation: small oscillations about the average are normal, we are looking out for large deviations of the average. Any such deviations indicate a pathology in the calculation and probably inaccurate results. In that case, you should analyse the output or trajectories to find out what is causing the problem and change calculation settings accordingly (different time step, thermostat, longer equilibration times, etc).

Part II: Dispersion Corrections

There are many methods which treat dispersion interactions within DFT. The simplest approach is to divide the molecular system into atomic fragments and employ a pairwise approximation. This leads to expressions of the form

$$\sum_{i<j} f(R_{ij}) \frac{C_{6,ij}}{R_{ij}^6} \tag{2}$$

where the sum runs over pairs of atoms i, j, R is the distance between said atoms and C_6 is a coefficient describing the strength of the interaction. f is a so-called damping function which is an empirical entity preventing R^{-6} from diverging and ensuring a smooth coupling of the dispersion energy to the DFT energy.

In the simplest variant, tabulated empirical C_6 coefficients for individual atoms are used and f has a simple analytical form (DFT-Dn series [4, 5]). In a more elaborated scheme, the $C_6[n]$ coefficients are functionals of the electron density n and thus take into account the local chemical environment (Tkatchenko–Scheffler [6]). In a yet more complex scheme, the C_6 coefficients may be derived from the Kohn–Sham orbitals (Becke–Johnson [7]).

An alternative approach to dispersion is the use of non-local density functionals of the form

$$\iint n(\mathbf{r}_1)\Phi[n](\mathbf{r}_1, \mathbf{r}_2)n(\mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 \tag{3}$$

where Φ is the so-called non-local kernel. That is also the quantity which distinguishes different non-local functionals. The two most widely adopted families are vdW-DF [8] and Vydrov–van Voorhis [9].

The non-local functionals share with the pairwise method that they are essentially two-body, or additive for long distances. But this oversimplifes the dispersion interaction picture. There are several ways to go beyond the pairwise approximations. The simplest approach is to extend the pairwise methods by adding three-body and higher-body terms. This is a strategy of the Many-Body Dispersion (MBD) method [10, 11], which relies on approximate response functions. The idea of MBD is to approximate response of atoms to electromagnetic field by harmonic dipole oscillators which can then be solved exactly. The key quantity in this model is the dynamic polarizability $\alpha(\omega)$ which is related to C_6 coefficients by the formula,

$$C_{6,ij} = \frac{3}{\pi}\int_0^\infty \alpha_i(\mathrm{i}\omega)\alpha_j(\mathrm{i}\omega)\mathrm{d}\omega \tag{4}$$

None of the methods mentioned above are exact and in this tutorial, we will limit ourselves to the Tkatchenko–Scheffler (TS) method [6] and the Many-Body Dispersion (MBD) [10, 11] method as representatives of the pairwise and many-body classes. They are both implemented in FHI-aims and well established.

Keep in mind that both TS and MBD methods have some shortcomings. Apart from the self-evident absence of many-body effects in the pairwise TS method, both methods suffer from the fact that they are formulated in terms of atoms. Hence in systems where the partitioning into atoms is problematic, such as in metals (delocalized electrons) or ionic systems (charge transfer), these methods may give inaccurate or even qualitatively wrong results. The obvious remedy might be to turn at least partly from atoms to electron density (cf. non-local functionals), and this framework is under development.

Different dispersion corrections in FHI-aims will be introduced in the example of a benzene molecule. Its aromatic ring features out-of-plane charge density that is readily polarised by dipole oscillations nearby. Consequently, binding of benzene to other materials is dominated by dispersion interactions.

Problem II: Pairwise Dispersion Correction

Educational Objectives:

- Find Hirshfeld volume and charge of carbon and hydrogen in a benzene molecule.
- Obtain C_6 coefficients calculated from short-range screened polarizability.

To compute pairwise dispersion corrections to DFT energy, the TS method employs C_6 coefficients that are calculated from the electron density. For this to be possible, the density has to be first divided between individual atoms. This can be done in many ways and FHI-aims utilizes the so-called Hirshfeld partitioning (the charge density partitions are also employed to compute partial charges). Once "atomic densities" are computed through partitioning, many quantities can be calculated. The system C_6 coefficients in Eq. (2) are obtained as:

$$C_6^{\text{eff}} = C_6^{\text{free}} \left(\frac{V_{\text{Hirshfeld}}}{V_{\text{free}}} \right)^2 \tag{5}$$

where C_6^{eff} is the effective atomic C_6 in a molecule and C_6^{free} is C_6 of a free atom. The volumes can be computed as stated in the Appendix.

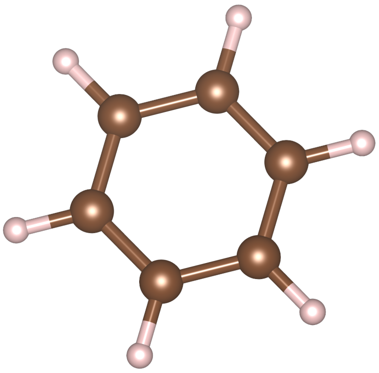


Figure 2: Single benzene molecule

We will now explore this in a calculation with the PBE functional and a reasonable SCF convergence criterion:

```
xc                  pbe
sc_accuracy_rho     1e-5
sc_accuracy_etot    1e-6
```

The following keyword selects the Tkatchenko-Scheffler (TS) method:

```
vdw_correction_hirshfeld
```

1. Create the `control.in` file for the calculation by listing only the keywords above. Append the basis set information to the control file, e.g. using:

   ```
   $ cat $SPECIES/light/*_{C,H}_* >> control.in
   ```

2. Create the geometry file for a benzene molecule. There are many alternative ways to do this, e.g. using tools like Avogadro or VESTA (in a cif format, and convert into `geometry.in` format using obabel). Here, we adopt a primitive but a powerful technique to create the benzene molecule using Octave, an open-source clone of MATLAB. We will do it by first creating a −CH element and then rotating it five times by 60°. We start with:

   ```
   $ octave --no-gui
   ```

   and enter the following commands:

   ```
   > rCC = 1.397
   > rCH = 1.084
   > CH = [rCC 0 0; rCC+rCH 0 0] # sets a 2-by-3 matrix
   > rot = [cosd(60) sind(60) 0; -sind(60) cosd(60) 0; 0 0 1]
   > benzene = CH;
   > for i = 1:5
   >     CH = CH*rot; # matrix multiplication of CH and rot
   >     benzene = [benzene; CH]; # appends rows of CH to rows of benzene
   > end
   > disp(benzene) # short for display
   ```

This will display the *xyz* coordinates for a benzene molecule where the C and H atoms are alternating. Save the coordinates to a file and exit from Octave (`Ctrl+D`). To convert the file into the correct format for `geometry.in` file, use the following `awk` script and visualise the structure to validate it.

```
$ awk '{print "atom",$0,(i++%2==0)?"C":"H"}' [file]
```

3. Run FHI-aims on 6-8 CPUs and output results to `TS-aims.out` file. Example with 6 CPUs:

   ```
   $ mpirun -n 6 aims.x > TS-aims.out
   ```

4. The TS calculation is run after the SCF cycle on the converged charge density, which is typical for dispersion corrections. Search for "Hirshfeld analysis" to find the following lines:

   ```
   Evaluating non-empirical van der Waals correction (Tkatchenko/Scheffler 2009).
   Performing Hirshfeld analysis of fragment charges and moments.
   ----------------------------------------------------------------------
   | Atom    1: C
   ```

5. Note the partial charges obtained by Hirschfeld partitioning. Which element tends to donate charge? Which part of the molecule is consequently charged?

6. Note the effective Hirshfeld volumes obtained by partitioning. How do they compare to the their free-atom counterparts?

7. Locate the TS energy correction in the line `vdW energy correction`, as seen below. The total DFT energy (without corrections) can now be computed by removing the dispersion correction from the total system energy.

```
Total energy components:
[...]
  | vdW energy correction      :         -0.00252031 Ha        -0.06858101 eV
[...]
  | Total energy               :        -232.0288263 Ha     -6313.82714026 eV
```

Problem III: Many-Body Dispersion Correction

Educational Objectives:

- Plot dynamic polarizabilities along the principal axes of the polarizability tensor.
- Compute the molecular C_6 coefficient from the dynamic polarizability.

In MBD, C_6 are computed via Eq. 4, and require as input the dynamic polarizability. Dynamic polarizability $\alpha(\omega)$ describes the response of a molecule in a similar way as static polarizability $\alpha(0)$ does, but for an oscillating electric field.

$$\mathbf{p}(\omega) = \boldsymbol{\alpha}(\omega)\mathbf{E}(\omega)$$

Here ω is the frequency, $E(\omega)$ is the generating field, $p(\omega)$ is the induced oscillating dipole moment and $\alpha(\omega)$ is the dynamic polarizability which is in general a tensor. As light is just an oscillating electric field from the point of view of many processes, the knowledge of dynamic polarizability is enough to calculate many interesting matter–light interaction properties. On the other hand, dispersion energy can be viewed as coming from fluctuations of these dipoles, which can be described by the same molecular polarizability (these two phenomena, inducing and fluctuating of the electric dipole in matter can be related by the fluctuation-dissipation theorem).

To activate the MBD method, replace `vdw_correction_hirshfeld` in the control file with

```
many_body_dispersion
```

Note that only one of these keywords should be used at the same time.

1. Run FHI-aims again and check the resulting `MBD-aims.out` file.

   ```
   $ mpirun -n 6 aims.x > MBD-aims.out
   ```

2. The MBD dispersion correction energy can be identified by the Libmbd routine, and can be found by searching for `Libmbd: Evaluated energy:` in the `MBD-aims.out` output. The total DFT energy without correction can be computed by removing the dispersion correction from the total system energy. Remember that the MBD energy is displayed in Ha! Check if the total DFT energy in this calculation is the same as the one in `TS-aims.out`

   ```
   Libmbd: Calculating MBD energy...
   Libmbd: Evaluated energy: -0.78925212322857874E-002
   ```

   ATTENTION!! Dynamic polarizabilities are not printed into the output by default anymore in version 210716.3 of FHI-aims. Further items of Problem III are not required unless you use an older version of FHI-aims. There is a option of printing these information in an extra output xml file but we haven't tested this option yet. Feel free to try it yourself if you know what you're doing ;)

3. Hirshfeld partitioning is performed again to compute partial charges, but it is not used to evaluate C_6 coefficients. The key input now is the dynamic polarizability, found in the lines:

   ```
   | Many-Body Dispersion (MBD@rsSCS) energy
   | Total dynamic anisotropic polarizability, \alpha_{ij}(iu), (a.u.)
   | -----------------------------------------------------------------
   |      u          xx         yy         zz         yz         xz         xy
   |   0.000000    82.950     82.955     36.363      0.000      0.000     -0.000
   |   0.003624    82.946     82.951     36.361      0.000      0.000     -0.000
   [...]
   ```

   In the output, the isotropic and principal components of $\boldsymbol{\alpha}(\omega)$ (ω is labeled as `u`) are listed. The benzene molecule orientation was chosen so that the principal axes coincide with the Cartesian axes. The dynamic polarizabilities have been screened, i.e. adjusted for the influence of neighboring atoms. The polarizabilities need to be screened only for the short-range part of the Coulomb potential, which is called "range-separated self-consistent screening", or "rsSCS".

4. Find the atom-projected static polarizability $\alpha(0)$ and the resulting MBD C_6 coefficients in the lines:

   ```
   | Partitioned atomic C6 coefficients and polarizabilities (a.u.)
   | -------------------------------------------------------------
   | ATOM   1  C   30.930690        9.015852
   | ATOM   2  H    1.870707        2.220960
   | ATOM   3  C   30.928155        9.015340
   | ATOM   4  H    1.872223        2.221900
   [...]
   ```

   "Partitioned" here does not mean actual partitioning of the molecular C_6 coefficient, but rather rescaling of free atom values using information from the Hirshfeld partitioning (see Appendix for details).

5. To analyse the dynamic polarizability $\alpha_i(\omega)$, isolate the listing (under `|   u   xx  ...`) and save it in `omega.txt`. Remove the leading pipe symbols in the polarizability output by using:

   ```
   $ sed -i 's/|//g' omega.txt
   ```

   Then start Octave again and type in the following commands to display the graph of dynamic polarizabilities. (`octave --no-gui`)

   ```
   > load omega.txt
   > alpha = omega(:, 2:4);
   > omega = omega(:, 1);
   > plot(omega, alpha);
   ```

```
> title("dynamic polarizablity")
> xlabel("frequency (a.u.)")
> ylabel("polarizability (a.u.)")
> legend("alpha_{xx}", "alpha_{yy}", "alpha_{zz}")
> xlim([0,6])
```

Note that the labels xx, yy and zz now refer to the principle axes of the polarizability tensor, and do not longer necessarily align with the cartesian axes x, y and z. The polarizability decreases with frequency as the electrons are not able to respond fast enough to rapid electromagnetic oscillations. Note the difference between the in-plane(alpha_xx and alpha_yy) and out-of-plane (alpha_zz) polarization - does this agree with what you know about the charge density distribution in benzene? At higher frequencies, this anisotropy disappears because the screening ceases to be effective.

6. Next, calculate the molecular C_6 coefficient from the dynamic polarizability using Eq. (4), where $\int \mathrm{d}\omega \rightarrow \sum_i w_i$. The integration weights are hard-coded in FHI-aims but a list is provided here for you. You can use the molecular C6 coefficient in Octave with (the same session as for the plotting):

```
> load int_weight.txt # load integration weight
> C6 = 3/pi*sum(alpha(:, 1).^2.*int_weight)
> # .^ and .* stand for elementwise operations
```

How large is the molecular C6 coefficient? Validate your results with the provided solution.

Part III: Intermolecular Dispersion Interactions

The importance of DFT dispersion corrections can be seen in intermolecular interactions. We continue with the benzene molecule and consider the molecule-molecule interaction as a function of molecular separation and orientation. In the subsequent problems we would like to compare TS and MBD energy contributions, but avoid running two separate calculations. MBD will be the main calculation performed, and we will use the mbd script supplied with this tutorial to additionally compute the TS energy contribution. By total energy *here* is the bare DFT energy (PBE in our case) *before* the final MBD@rsSCS correction.

Educational Objectives

- Compute the total energy minimum and the intermolecular distance for a stacked benzene dimer.
- Compare the predictions of PBE, PBE+TS and PBE+MBD with a reference method.

Problem IV: Stacked benzene interaction

The interaction curve of two molecules A and B is:

$$E_{\text{int}}(r) = E(AB, |AB| = r) - E(A) - E(B)$$

where $E_{\text{int}}(r)$ is the interaction energy of molecules at a distance r and E are total energies of the independent systems. In the case of a benzene dimer, $A = B$ so we need to run FHI-aims for a single benzene molecule once. Then we compute a series of geometries with parallel benzene dimers at different separations. The provided Python script `run.py` creates all geometries and performs all these calculations sequentially. The resulting calculations can be analysed using the provided Python script `extract.py`.

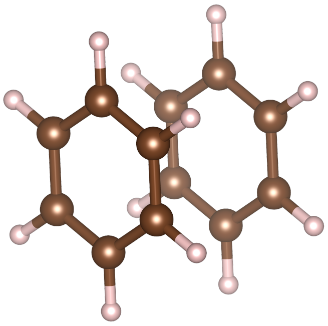


Figure 3: Stacked benzene dimer

1. Copy over all template files. Check the template `control.in` file. The basis set will be added by the script `run.py` automatically.

2. In `run.py`, consider the `# build benzene molecule` and `# loop over distances` sections. Can you relate them to the previous exercise and the task above?

3. Run the script on 6 CPUs by typing `python run.py 6`. Enter the resulting directories and visualise the `geometry.in` to verify it.

4. Now we employ a script to compute the TS and MBD dispersion contributions. For this we have to suspend the usual computational modules. Use the `module purge` command, then run the script by typing `python3m get_mbd.py 6`. Check that the TS dispersion correction (that was computed separately) was correctly written into the `ts.out` file. Now load the modules again by typing `module load intel/psxe-2020` (you can check if the change took place by typing `module list`).

5. Run the command `sed -i '1d' RUN.1/*/ts.out` . The script `extract.py` will initially gather the data and calculate the interaction energies for different methods: `DFT energy`, `Total energy + TS energy`, and `Total energy + MBD@rsSCS energy`. Run the script by typing `python3m extract.py`.

6. The results are plotted alongside a reference coupled cluster (CCSD(T)) value. How does pure PBE perform for this system? Which dispersion correction performs better here?

Problem V: Investigation of the benzene dimer PES

The benzene dimer has served as a test case for dispersion methods since the 1980s and it took over two decades to finally resolve its potential energy surface (PES). Particularly challenging is the description of almost degenerate T-shaped (T) and parallel-displaced (PD) structures.

We will reproduce the main features of the PES by computing a series of stable dimer configurations that feature different molecule-molecule distances and orientations. These geometries represent stationary points on the 3D potential energy surface (PES) of the benzene dimer. As before, all calculations will be run sequentially by the `run.py` script, and analysed by the `extract.py` script.

1. Copy over all template files. Inspect the `bz-dimer.geoms.txt` file that will serve as input for `run.py`, and visualise the related xyz file. How many geometries will be computed?

   ```
   M1
   24

   C     1.205074   -0.695750    0.000000
   # ...
   H    -2.140382   -0.474604    3.488382
   ------------------------------------------------------------
   M2
   24
   # ...
   ```

   You can visualize all geometries with `jmol bz-dimer.geoms.xyz` . Use the arrows in the gui to list through the configurations.

2. Execute `run.py` as before on 6 CPUs by typing `python3m run.py 6`. Enter the relevant directories to make sure that all calculations complete successfully.

3. Again, we employ a script to compute the TS and MBD dispersion contributions. Use the `module purge` command, then run the script by typing `python get_mbd.py 6`. Load the modules again by typing `module load intel/psxe-2020`.

4. Run the command `sed -i ’1d’ RUN.1/*/ts.out` . Inspect the file `bz-dimer.energies.csv` that will serve as comparison for our calculations, then run `python3m extract.py` script as before. How do the different methods perform overall? Which method overbinds, which one underbinds and why?

5. Compare the reference energies against the xyz structures: which dimer configuration is lowest in energy? Does the most accurate DFT method also predict this configuration to be lowest in energy?

6. The most difficult problem in describing the benzene dimer PES is the comparison of T-shaped (T) and parallel-displaced (PD) structures: they should be almost degenerate ($< 0.1\,\text{kcal/mol}$). Do our calculations correctly reproduce this result?

Appendix I: Theoretical background

Here, we would like to present a brief description of the Many-Body Dispersion (MBD) method. The procedure starts with projecting the response of valence electrons onto a set of quantum harmonic oscillators by calculating the “volumes” for free atoms and atoms in a molecule,

$$V_i = \int n_i(\mathbf{r})|\mathbf{r} - \mathbf{R}_i|^3 \mathrm{d}\mathbf{r}$$

where $\mathbf{R}_i$ is the position of atom i and n_i is either its free-atom electron density n_i^{free} or Hirshfeld density n_i^{H},

$$n_i^{\text{H}}(\mathbf{r}) = n(\mathbf{r}) \frac{n_i^{\text{free}}(\mathbf{r})}{\sum_j n_j^{\text{free}}(\mathbf{r})}$$

The ratio of the free and Hirshfeld volumes captures the influence of chemical environment on the individual atoms. To incorporate this effect in the dispersion interaction, the free atom reference static polarizabilities α_0, C_6 coefficients and dispersion radii R^{vdW} are renormalized using the volume ratios,

$$\begin{aligned} \alpha_{0,i} &= \alpha_{0,i}^{\text{free}} \times V_i^{\text{H}}/V_i^{\text{free}} \\ C_{6,i} &= C_{6,i}^{\text{free}} \times (V_i^{\text{H}}/V_i^{\text{free}})^2 \\ R_i^{\text{vdW}} &= R_i^{\text{vdW,free}} \times (V_i^{\text{H}}/V_i^{\text{free}})^{1/3} \end{aligned}$$

Then, the dynamic polarizabilities are calculated as

$$\alpha_i(\omega) = \alpha_{0,i} \left(1 - \frac{\omega^2}{(\omega_i^{\text{eff}})^2}\right)^{-1}$$

where the effective resonance frequency ω_i^{eff} is defined as

$$\omega_i^{\text{eff}} = \frac{4C_{6,i}}{3\alpha_{0,i}^2}$$

This corresponds to approximating the response of an atom with a single harmonic oscillator.

Screening In the next step, we evaluate the screening of polarizabilities, i. e. how response properties of the atoms are influenced by other atoms. The scalar polarizabilities are first recast as tensors, $\boldsymbol{\alpha}_i = \boldsymbol{\delta}\alpha_i$ ($\boldsymbol{\delta}$ for Kronecker delta) and gathered into one $3N$-by-$3N$ matrix, where N is the number of atoms, $\boldsymbol{\alpha} = \bigoplus_i \boldsymbol{\alpha}_i$. This is the bare polarizability tensor $\boldsymbol{\alpha}$. Next, we want to let the atoms interact through dipole interaction,

$$\mathbf{T}_{\text{dipole}} = \boldsymbol{\nabla} \otimes \boldsymbol{\nabla} \frac{1}{r}$$

To explicitly account for the fact that we do not have point dipoles, but rather atoms where the dipole density is spread over a finite volume, we consider the Coulomb interaction between two Gaussian charge densities, $v_{GG}(r) = \operatorname{erf}(r/\sigma)/r$. Furthermore, we want to perform the screening only with the short-range part of the Coulomb potential. This leads to the final interaction tensor between atoms i and j of the form

$$\mathbf{T}_{ij}^{GG,\text{sr}} = (1 - f(R_{ij}))\, \boldsymbol{\nabla} \otimes \boldsymbol{\nabla} \frac{\operatorname{erf}(r/\sigma_{ij})}{r}$$

where $\sigma_{ij} = \sqrt{\sigma_i^2 + \sigma_j^2}$, $\sigma_i = \left(\sqrt{2/\pi} \times \alpha_i/3\right)^{1/3}$ defines the natural width of the interacting charge densities, f is a damping function and R_{ij} is the distance between the atoms.

The short-range screened polarizability tensor $\boldsymbol{\alpha}^{\mathrm{sr}}$ is then obtained using the self-consistent Dyson-like equation,

$$\boldsymbol{\alpha}^{\mathrm{sr}} = \boldsymbol{\alpha} + \boldsymbol{\alpha}\mathbf{T}^{GG,\mathrm{sr}}\boldsymbol{\alpha}^{\mathrm{sr}}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \dots & \mathbf{T}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{T}_{N1} & \dots & \mathbf{T}_{NN} \end{bmatrix}$$

The isotropic atomic screened polarizabilities are recovered as

$$\alpha_i^{\mathrm{sr}}(\omega) = \frac{1}{3}\operatorname{Tr}\left[\sum_j \boldsymbol{\alpha}_{ij}^{\mathrm{sr}}(\omega)\right]$$

RPA Finally in the last step, the MBD correlation energy is calculated from the short-range screened polarizabilities (response functions) using the random-phase approximation (RPA) formula,

$$E_{\mathrm{c}} = \frac{1}{2\pi}\int_0^\infty \operatorname{Tr}[\ln(1-\chi_0 v) + \chi_0 v]\mathrm{d}\omega$$

where in case of MBD, the bare response function χ_0 is $\bigoplus_i \boldsymbol{\delta}\alpha_i^{\mathrm{sr}}$ and the interaction potential v is ("lr" for long-range)

$$\mathbf{T}^{\mathrm{lr}} \sim \mathbf{T}_{ij}^{\mathrm{lr}} = f(R_{ij})\boldsymbol{\nabla}\otimes\boldsymbol{\nabla}\frac{1}{r}$$

A significant convenience of MBD is that because of the use of dipole approximation and other considerations, the RPA integral can be evaluated analytically by recasting the problem as matrix diagonalization.

Periodic MBD

The MBD method could be of course made periodic using Bloch functions and k-point sampling, but we employ a less technically demanding solution. In the screening part, the effect of atoms from other unit cells is incorporated in the interaction tensor $\mathbf{T}$ by

$$\mathbf{T}_{ij} = \sum_k \mathbf{T}_{ik}, \qquad k \in \{\text{replicas of } j, |jk| < R_{\text{cutoff}}\}$$

In FHI-aims, R_{cutoff} can be controlled with the keyword `mbd_scs_dip_cutoff`.

In the RPA part, the modification is two-fold. First, the interaction tensor is modified in the same way as for screening, just using a different keyword `mbd_cfdm_dip_cutoff`. Second, the original unit cell is replicated into a supercell, the RPA integral is evaluated for the whole supercell as if it was a molecular cluster, and the MBD energy is averaged over all unit cells that constitute the supercell. The size of the supercell is controlled with `mbd_supercell_cutoff`. Mind that you should always check the convergence of MBD energy with all these three parameters in production periodic calculations.

References

[1] I. Newton, *Philosophiæ Naturalis Principia Mathematica*, London (1687).

[2] Dominik Marx et al. “The nature of the hydrated excess proton in water”. In: *Nature* 397 (Feb. 1999), pp. 601–604. ISSN: 0028-0836. DOI: 10.1038/17579. URL: http://dx.doi.org/10.1038/17579.

[3] D. Frenkel and B. Smit, *Understanding Molecular Simulation: from algorithms to applications*, second edition, Academic Press 2002.

[4] Stefan Grimme. “Accurate description of van der Waals complexes by density functional theory including empirical corrections”. In: *Journal of computational chemistry* 25.12 (2004), pp. 1463–73. ISSN: 0192-8651. DOI: 10.1002/jcc.20078. URL: http://www.ncbi.nlm.nih.gov/pubmed/15224390.

[5] Stefan Grimme et al. “A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu”. In: *The Journal of Chemical Physics* 132.15 (2010), p. 154104. DOI: 10.1063/1.3382344.

[6] Alexandre Tkatchenko and Matthias Scheffler. “Accurate Molecular Van Der Waals Interactions from Ground-State Electron Density and Free-Atom Reference Data”. In: *Physical Review Letters* 102.7 (2009), p. 073005. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.102.073005. URL: http://link.aps.org/doi/10.1103/PhysRevLett.102.073005.

[7] Axel D. Becke and Erin R. Johnson. “A density-functional model of the dispersion interaction”. In: *The Journal of Chemical Physics* 123.15 (2005), p. 154101. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.2065267. URL: http://scitation.aip.org/content/aip/journal/jcp/123/15/10.1063/1.2065267.

[8] M. Dion et al. “Van der Waals Density Functional for General Geometries”. In: *Physical Review Letters* 92.24 (2004), p. 246401. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.92.246401. URL: http://link.aps.org/doi/10.1103/PhysRevLett.92.246401.

[9] Oleg A. Vydrov and Troy Van Voorhis. “Nonlocal van der Waals density functional: The simpler the better”. In: *The Journal of chemical physics* 133.24 (2010), p. 244103. ISSN: 1089-7690. DOI: 10.1063/1.3521275. URL: http://www.ncbi.nlm.nih.gov/pubmed/21197972.

[10] Alberto Ambrosetti et al. “Long-range correlation energy calculated from coupled atomic response functions”. In: *The Journal of Chemical Physics* 140.18 (2014), 18A508. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.4865104. URL: http://scitation.aip.org/content/aip/journal/jcp/140/18/10.1063/1.4865104.

[11] Robert A. DiStasio Jr., Vivekanand V. Gobre, and Alexandre Tkatchenko. “Many-body van der Waals interactions in molecules and condensed matter”. In: *Journal of Physics: Condensed Matter* 26.21 (2014), p. 213202. ISSN: 0953-8984. DOI: 10.1088/0953-8984/26/21/213202. URL: http://iopscience.iop.org/0953-8984/26/21/213202.