

Information about gen_dipole.f90

M. A. Seaton and S. Chiacchiera

August 23, 2017

This module, `gen_dipole.f90`, is a generalization of the `dipole.f90` post-processing utility of DL_MESO_DPDP, the Dissipative Particle Dynamics (DPD) code from the DL_MESO package [1, 2]. The module is to be used with DL_MESO in its last released version, version 2.6 (dating November 2015).

It processes the trajectory (HISTORY) files to obtain the charge dipole moments of all the (neutral) molecules in the system. It produces files `dipole_*` containing the time evolution of relevant quantities (see below). In the case of a single molecular species, it also prints to the standard output the Kirkwood number g_k and the relative electric permittivity ϵ_r for this species, together with an estimate for their errors (standard error).

The module can be applied to systems including molecules with a generic charge structure, as long as each molecule is neutral (otherwise the charge dipole moment would be frame-dependent).

CAVEAT: this module only analyzes molecular trajectories. If a charged molecule is present, an error message will be given, while unbonded charges would not be detected and would lead to erroneous results. Therefore please keep in mind **to not apply** the module to systems with unbonded charges.

1 Disambiguation on the concept of molecule

In DL_MESO a *defined molecule* is a set of beads, which can be bonded or not.

For the purpose of this module it is *required* that each molecule is a connected cluster (via stretching bonds). In fact, this, together with the reasonable assumption that each stretching bond cannot be stretched to more than half the system linear size, allows to univocally define the charge dipole moment of each molecule.

2 Charge dipole moments

The charge dipole moment of a neutral molecule is

$$\vec{p}_{mol} = \sum_{i \in mol} q_i \vec{r}_i \quad (1)$$

where \vec{r}_i are the bead positions and q_i their charges. The total charge dipole moment of the simulated volume V is $\vec{P} = \sum_{mol \in V} \vec{p}_{mol}$. If two or more molecular species are present, one can split \vec{P} into the different species contributions.

The Kirkwood number for a pure system is

$$g_k = \frac{\langle \vec{P}^2 \rangle}{N_{mol} \langle \vec{p}^2 \rangle}, \quad (2)$$

where $\langle \dots \rangle$ indicates an average over trajectories. If the dipoles orientations are not correlated, then $g_k \simeq 1$.

The relative dielectric permittivity of the medium is calculated from linear response theory

$$\epsilon_r = 1 + \frac{4\pi}{3} l_B \frac{\langle \vec{P}^2 \rangle}{V}, \quad (3)$$

where l_B is Bjerrum length, V the system volume, and tin-foil boundary conditions are used.

3 Estimate of errors

The errors on $\langle \vec{P}^2 \rangle$ and on $\langle \vec{p}^2 \rangle$ are estimated assuming that each snapshot is an independent measure. Averages are computed as

$$\langle A \rangle = \sum_{it=1}^{N_{av}} A_{it} \quad (4)$$

where A_{it} refers to a snapshot, and N_{av} is the total number of snapshots in the HISTORY* file(s).

The errors are computed as:

$$\delta \langle A \rangle = \frac{\sigma_A}{\sqrt{N_{av}}} \quad (5)$$

where

$$\sigma_A^2 = \langle (A - \langle A \rangle)^2 \rangle = \langle A^2 \rangle - \langle A \rangle^2 . \quad (6)$$

For derived quantities (as g_k) the errors are propagated assuming they are not independent of each other:

$$\frac{\delta g_k}{g_k} = \frac{\delta \langle \vec{P}^2 \rangle}{\langle \vec{P}^2 \rangle} + \frac{\delta \langle \vec{p}^2 \rangle}{\langle \vec{p}^2 \rangle} \quad (7)$$

and

$$\delta \epsilon_r = \frac{4\pi}{3} l_B \frac{\delta \langle \vec{P}^2 \rangle}{V} . \quad (8)$$

4 Compiling

The present module `gen_dipole.f90` is compiled with the available Fortran90 compiler, e.g.:

```
gfortran -o gen_dipole.exe gen_dipole.f90
```

and the executable must be in the same directory of the HISTORY* files to be analyzed.

5 Input

The user is asked to provide the number of nodes used to run the simulation, the electric charges for all the types of beads and the Bjerrum length.

To input these parameters one can enter them from the keyboard or write them into a text file (say, `input.txt`), one per line (in the right order) and run the program in this way:

```
gen_dipole.exe < input.txt
```

6 Output

In any case, the average quantities

$$\begin{aligned} &\langle P_x \rangle, \langle P_y \rangle, \langle P_z \rangle \\ &\langle \frac{\vec{P}^2}{V} \rangle \\ &\langle \frac{\sum_{i=1}^{N_{mol}} \vec{p}_i^2}{N_{mol}} \rangle \end{aligned}$$

and their errors are printed to the standard output. When more than one molecular species are present, for example two, the order in which they are printed is

- $\langle P_x \rangle_{motype1}, \langle P_x \rangle_{motype2}, \langle P_y \rangle_{motype1}, \langle P_y \rangle_{motype2}, \langle P_z \rangle_{motype1}, \langle P_z \rangle_{motype2}$

and so on.

If *only one* molecular species is present, the module prints to the standard output g_k and ϵ_r for this species, together with their error estimates. Otherwise, no value is provided for these quantities.

6.1 Output file(s)

For each molecular species, a file called `dipole_*` is produced and contains:

- snapshot index, $P_x, P_y, P_z, \sum_{i=1}^{N_{mol}} \frac{\vec{p}_i^2}{N_{mol}}, \frac{\vec{P}^2}{V}$.

7 Checks

- **Order of beads' labels.** The module checks that the beads are labelled in the following way:
 - Fix a molecule type ($i = 1, \dots, nmoldef$)
 - Fix a molecule index ($j = 1, \dots, nmol(i)$)
 - Beads of the molecule ($k = 1, \dots, nbdmol(i)$)
 - **Molecule content.** It checks that for a given molecular type the bead content of all the molecules is the same (and in the same order).
 - **Molecule charge neutrality.** An independent check is done on charges: it is checked that all molecules are charge neutral.
 - **Connectivity.** Within `compute_charge_dipoles` it is verified that each molecule is a connected cluster (see next section).

7.1 Subroutine connect

Subroutine: `connect` Arguments: `nbeads, nbonds, bndtbl, visit, from`

<code>nbeads</code>	input	integer		number of beads in the system
<code>nbonds</code>	input	integer		number of bonds in the system
<code>bndtbl</code>	input	integer	A (nbonds,2)	with extremes of all bonds
<code>visit</code>	output	integer	A (nbeads)	order to visit following bonds
<code>from</code>	output	integer	A nbeads	reference beads for <code>visit</code>

Internally, this subroutine evaluates different properties of clusters, which include: the *degree* of each bead (i.e., the number of links having it as an extreme), the *nearest neighbours list* and a histogram of clusters sizes (i.e., the number of beads they contain).

NB: Within this subroutine it is assumed that beads' labels are ordered into connected blocks. In other words, if it happens that there are clusters containing $\{1, 2, 4\}$ and $\{3, 5\}$ an error will be detected. The reason for this assumption is that it simplifies the detection of clusters.

8 Test

We propose two tests to familiarize with the utility and a third one on a physically relevant system.

The first two tests involve two (toy) molecular species: a branched one (four beads, T-shaped) and a simple dimer. All the beads carry charges. In the first case 10 molecules of each type are present and are

species	A	B	C	D
charge	0.2	-1	0.6	1

Table 1: Charges

followed for a few time steps. In the second case it is suggested to analyze a single snapshot with just two molecules and all the beads sitting at user-defined positions (via the CONFIG file).

Four type of beads are used (see Table 1) and the Bjerrum length is fixed as $l_B = 1$.

The bonding connections in the two molecules are pictorially given in Fig. (1) below:

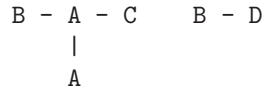


Figure 1: Molecules in the test: beads and bonds.

8.1 Test 1

Run DL_MESO_DPD on a single core (serial run) using for the CONTROL file

```

-----
volume 3.0 3.0 3.0
temperature 1.0
cutoff 1.0

timestep 0.01
steps 1000
equilibration steps 0
traj 0 100 0
stats every 100
stack size 100
print every 100
job time 1000.0
close time 10.0

ensemble nvt mdvv
conf origin zero

ewald sum 1.0 5 5 5
bjerrum 1.0
smear gauss equal

finish
-----

```

and for the FIELD file

```

-----
Two kinds of molecules: branched and dimer

SPECIES 4
A 1.0 0.2 0 0
B 1.0 -1.0 0 0
C 1.0 0.6 0 0
D 1.0 1.0 0 0

MOLECULES 2
BRANCH
nummols 10
beads 4
B 0.0 0.0 0.0
A 0.0 0.2 0.0
C 0.0 0.4 0.0
A 0.2 0.2 0.0
bonds 3
harm 1 2 5.0 0.25
harm 2 3 5.0 0.25
-----

```

```

harm 2 4 5.0 0.25
finish
BD
nummols 10
beads 2
B 0.0 0.0 0.3
D 0.0 0.0 0.1
bonds 1
harm 1 2 5.0 0.25
finish

```

```

INTERACTIONS 4
A A dpd 25.0 1.0 4.5
B B dpd 25.0 1.0 4.5
C C dpd 25.0 1.0 4.5
D D dpd 25.0 1.0 4.5

```

CLOSE

Analyzing the HISTORY files with gen_dipole.exe this output is printed on the standard output

```

-----
Number of nodes used in calculations ?
nchist:      0      10      0      10      0      0      0      0      0      0
Charges for SPECIES type A      :
Charges for SPECIES type B      :
Charges for SPECIES type C      :
Charges for SPECIES type D      :
chg=      0.2000     -1.0000      0.6000      1.0000
Number of snapshots:      11
<P_x>, <P_y>, <P_z>:
1.004169E+00  2.898942E-01  -1.434444E+00  -6.251265E-01  1.620951E-01  4.372775E-01
error:
7.907981E-01  3.949007E-01  7.273669E-01  2.906167E-01  3.410938E-01  4.811729E-01
<P^2>/V:
6.322489E-01  2.169365E-01
error:
2.328649E-01  6.785635E-02
<p^2>:
1.319790E+00  8.467037E-01
error:
1.315487E-01  1.129629E-01
-----

```

The first line shows the histogram of cluster sizes: in this case, it correctly gives 10 molecules of two beads, and 10 molecules of 4 beads. Since internally the module checks that each molecule is a connected cluster, this line should always give a histogram with the molecule sizes (by default, shown up to ten beads).

The dipole_DB file is:

```

-----
 1  5.610985E-02  1.656239E-01  -1.397198E-01  4.000000E-02  1.855601E-03
 2  4.642350E-01  -9.118123E-01  9.524455E-02  1.599716E+00  3.911064E-02
 3  -2.027409E+00  -1.131613E+00  3.758015E-01  4.747804E-01  2.048949E-01
 4  1.319446E+00  1.476795E-01  1.063918E+00  8.070183E-01  1.072100E-01
 5  2.010414E+00  -4.817778E-01  2.350176E+00  9.949271E-01  3.628594E-01
 6  -1.451075E+00  9.994575E-01  3.334998E+00  1.227528E+00  5.269164E-01
 7  -2.657225E-01  -7.582258E-01  2.300250E-01  7.640699E-01  2.586764E-02
 8  5.496076E-01  -8.682396E-01  1.023472E+00  8.465722E-01  7.790383E-02
 9  1.088395E+00  -2.984139E-01  -2.146073E+00  9.144278E-01  2.177513E-01
10  -8.548484E-01  -7.038363E-01  9.717777E-01  7.643206E-01  8.038901E-02
11  2.299682E+00  -3.035233E+00  -2.349568E+00  8.803798E-01  7.415426E-01
-----

```

and the dipole_BRANCH file is:

```

-----
 1  2.869309E-02  -4.152495E-01  1.999305E-01  1.040000E-01  7.897321E-03
 2  -1.875110E+00  -9.240273E-01  -1.062192E+00  1.335149E+00  2.036340E-01
 3  1.333150E+00  2.371575E+00  4.829357E-02  1.537261E+00  2.742218E-01
 4  -7.447839E-02  7.132180E-01  -5.185719E-01  1.806593E+00  2.900532E-02
 5  8.163357E+00  -2.563583E+00  -2.268061E-01  1.699112E+00  2.713474E+00
 6  1.909976E+00  -2.072637E+00  1.328292E+00  1.408593E+00  3.595626E-01
 7  -1.606210E+00  -3.637945E+00  1.086719E+00  1.279568E+00  6.294633E-01
 8  -7.687367E-01  -6.351486E+00  1.252305E+00  1.355228E+00  1.574096E+00
 9  1.428639E-01  -3.620968E+00  3.834862E-01  1.132792E+00  4.918104E-01
-----

```

```

10  2.524788E+00  1.276311E+00  1.641325E+00  1.187329E+00  3.962026E-01
11  1.267564E+00 -5.540963E-01 -2.349736E+00  1.672068E+00  2.753703E-01
-----

```

If instead the simulation is run on multiple nodes, only the results for the first snapshot will be unchanged (i.e., first line of each `dipole_*` file), the other results will vary because a different sequence of random numbers will enter the time evolution of the system.

8.2 Test 2

Run `DL_MESO_DPD` using the same `CONTROL` and `FIELD` files as above, with the only changes:

- "`steps 1000`" changes into "`steps 1`"
- "`nummols 10`" changes into "`nummols 1`" (NB: appears twice)

Also, use this `CONFIG` file that will initially put the molecules branches aligned with the cartesian axes

```

-----
Two kinds of molecules: branched and dimer
0  1
3.0  0.0  0.0
0.0  3.0  0.0
0.0  0.0  3.0
B  1
0.0 0.0 0.0
A  2
0.0 0.2 0.0
C  3
0.0 0.4 0.0
A  4
0.2 0.2 0.0
B  5
0.0 0.0 0.3
D  6
0.0 0.0 0.1
-----

```

where the identity of each bead is fixed by the `FIELD` file and is shown in Fig. (2) below

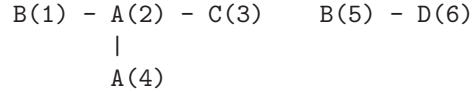


Figure 2: Beads with global labels and bonds.

One can easily check that the dipole of each molecule is as expected:

$$\vec{p}_{BRANCH} = (0.04, 0.32, 0), \quad \vec{p}_{BD} = (0, 0, -0.2) . \tag{9}$$

In fact: the `dipole_BRANCH` file is

```

-----
1  0.000000E+00  0.000000E+00 -2.000000E-01  4.000000E-02  1.481481E-03
-----

```

and the `dipole_BD` file is

```

-----
1  4.000000E-02  3.200000E-01  0.000000E+00  1.040000E-01  3.851852E-03
-----

```

The results of this test do not depend on the number of nodes used to run the simulation.

8.3 Test 3: water in oil

Here we suggest to consider a fluid made of harmonically bonded dimers $(+q, -q)$. Fixing appropriately the partial charge q and the Bjerrum length l_B this system mimics water in an oil background, as long as the dielectric properties are concerned. For more details about this model, please see Section 9.

Run DL_MESO_DPD using for the CONTROL file

```
-----  
DL_MESO charged harmonic dimers with dpd repulsion
```

```
volume 64.0  
temperature 1.0  
cutoff 1.0  
  
timestep 0.01  
steps 70000  
equilibration steps 20000  
traj 20000 100  
stats every 100  
stack size 100  
print every 100  
job time 7200.0  
close time 100.0
```

```
ensemble nvt mdvv  
  
ewald sum 1.0 5 5 5  
bjerrum 42.0  
smear gauss equal
```

```
finish  
-----
```

and for the FIELD file

```
-----  
DL_MESO charged harmonic dimers with dpd repulsion
```

```
SPECIES 2  
solp 1.0 0.46 0  
solm 1.0 -0.46 0
```

```
MOLECULES 1  
DIMER  
nummols 96  
beads 2  
solp 0.0 0.0 0.0  
solm 0.1 0.0 0.0  
bonds 1  
harm 1 2 5.0 0.0
```

```
finish
```

```
INTERACTIONS 3  
solp solp dpd 25.0 1.0 4.5  
solm solm dpd 25.0 1.0 4.5  
solp solm dpd 25.0 1.0 4.5
```

```
CLOSE  
-----
```

Analyzing the HISTORY file with `gen_dipole.exe`, this output is written on the standard output

```
-----  
Number of nodes used in calculations ?  
nchist: 0 96 0 0 0 0 0 0 0 0  
Charges for SPECIES type solp :  
Charges for SPECIES type solm :  
chg= 0.4600 -0.4600  
Number of snapshots: 501  
<P_x>, <P_y>, <P_z>:  
-3.627184E-02 -1.790324E-01 5.466988E-02  
error:  
9.792086E-02 1.066997E-01 9.685359E-02  
<P^2>/V:  
2.381825E-01
```

```

error:
8.641061E-03
<p^2>:
1.419998E-01
error:
4.556992E-04
kirkwood factor:
1.118229E+00
error:
4.415698E-02
Bjerrum length?
epsilon_r:
4.290325E+01
error:
1.520215E+00

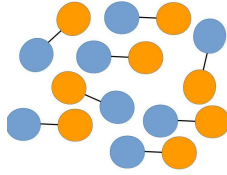
```

In particular, we see that:

- $\vec{P} = (0.0 \pm 0.1, 0.2 \pm 0.1, 0.1 \pm 0.1)$
- $\epsilon_r = 43 \pm 2$
- $g_k = 1.12 \pm 0.04$

Please notice that the error estimates are done assuming all the samples are independent. From the results obtained in the testing case of the module `gen_dipoleaf.f90`, one sees that the auto-correlation time of \vec{P} in this system is about 1-2 DPD time units, so the sampling choice done here (trajectories are written every 100 time steps, i.e., every 1 DPD time units) seems reasonable, even if a bit optimistic¹. To confirm the reliability of the error estimate one can do another run with a different random number sequence (see the directive `seed` of `DL_MESO`) and see if the two results are compatible within error bars.

9 Test case: a dimer solvent



Here we describe a physical system in which the software modules dealing with charge dipole moments in DPD simulations can be tested. It is a *polarizable fluid* made of harmonically bonded dimers ($+q, -q$), pictorially represented above (not in scale). Fixing appropriately the partial charge q and the Bjerrum length l_B , this system mimics *water in an oil background*, as long as the dielectric properties are concerned.

We recall that the electric permittivity is ϵ_0 for vacuum, and $\epsilon < \epsilon_0$ for a medium. The medium effect can be split into a background and a relative term $\epsilon/\epsilon_0 = \epsilon_b \epsilon_r$. The background is constant and uniform, whereas the explicit term is due to dynamic microscopic objects (dimers in this case) which carry a charge dipole moment. The strength of electrostatic interactions in a background is set by the bare Bjerrum length $l_B = e^2/(4\pi\epsilon_0\epsilon_b k_B T)$. On the other hand, from linear response theory, the bulk value of the relative permittivity is $\epsilon_r = 1 + \frac{\langle \vec{P}^2 \rangle_{\vec{E}=\vec{0}}}{3\epsilon_0\epsilon_b k_B T V}$, where tin-foil boundary conditions are assumed.

Two types of beads are present in the simulation, `solp` and `solm`, the solvent positive and negative partial charges, respectively. We fix the bare Bjerrum length $l_B = 42$ (appropriate for oil²), the repulsion parameter

¹To be on the safe side with the error estimate, one has to run long simulations and sample the trajectories at time intervals that are sufficiently long to have uncorrelated (for the variable of interest) samples.

²Notice that the physical length scale is set choosing r_c : if we choose $r_c = 0.646nm$ (appropriate to match water density at room temperature if $N_m = 3$, i.e., one bead represents three water molecules), the Bjerrum length of oil in DPD units is $l_B = 27nm \simeq 42r_c$, hence the value given above for the oil background.

$A = 25$, the harmonic spring constant $k = 5$, the bead density $\rho = 3$, the partial charges $|q| = 0.46$ and the Gaussian smearing length $\sigma = 0.5$. All quantities are given in DPD units, where $k_B T = 1$, $r_c = 1$ and $m = 1$. This fluid has a relative permittivity $\epsilon_r \simeq 40$, as can be checked using the `gen_dipole.f90` utility. This value is compatible with the ratio of water and oil permittivities $\epsilon^{water}/\epsilon^{oil} \simeq 40$.

References

- [1] www.ccp5.ac.uk/DL_MESO
- [2] M. A. Seaton, R. L. Anderson, S. Metz and W. Smith, *Mol. Sim.* **39** (10), 796 (2013).