Information about check_config.f90

M. A. Seaton and S. Chiacchiera

February 23, 2018

This module, `check_config.f90`, is a pre-processing utility for DL_MESO_DPD, the Dissipative Particle Dynamics (DPD) code from the DL_MESO package [1, 2]. It checks that the content of the optional configuration (CONFIG) file is consistent with that of the necessary input files (CONTROL and FIELD). In particular, it checks: the system dimensions, its composition and the bead content of all the molecules. In addition, in case hard walls are present, it checks that none of the stretching bonds between beads crosses a hard wall.

This module is to be used with DL_MESO in its last released version, version 2.6 (dating November 2015).

1 Input files content

The whole simulated volume (*system*) can be obtained replicating a building block (*unit cell*) along the Cartesian axes (directive `nfold`, in the CONTROL file). Another concept, that appears in the MPI version of the code, is a portion of the simulated volume assigned to a given process (*domain*). Since the present module is a serial one, the domain and system coincide.

Here we will focus on only some aspects of the information contained in the input files, namely: the system sizes, species populations, molecular bead content, beads positions and stretching bonds. Keeping this restriction in mind, we recall the quantities contained in each file:

- CONTROL (mandatory): system size and possibly duplication factors (`nfold`).
- FIELD (mandatory): population of unbonded species, molecule definition and population. It refers to the unit cell, its content is rescaled according to `nfold`.
- CONFIG (optional): can be used to define the system initial configuration (coordinates, and, optionally, velocities and forces). It also refers to the unit cell, its content is rescaled according to `nfold`. For each bead, a global index and the bead species are given.

2 Checks performed

The present module checks that

- the system volume is uniquely defined, either in the CONFIG file (possibly rescaled using `nfold`) or in the CONTROL file.
- the populations of the various species in the CONFIG file coincide with what defined in the FIELD file (possibly rescaled using `nfold`).
- the labeling of the molecular beads in the CONFIG file is done as expected [3], namely: following the ordering of molecule types and molecular beads defined in the FIELD file.
- if hard walls are present, it checks that no stretching bond is located across a hard wall.

- a warning is given signaling any bead coordinate out of the expected space region. (However, these coordinates are automatically corrected by DL_MESO_DPD).

3 Bead labeling within DL_MESO_DPD

To illustrate the bead labeling, we consider a simple example.

If the FIELD file is

```
----------------------------------------------------
Simple test

SPECIES 3
A  1.0  0.0  0
B  1.0  0.0  6
C  1.0  0.0  0

MOLECULES 2
ACB
nummols 4
beads 3
A  0.0 0.5 0.0
C  0.0 0.0 0.0
B  0.5 0.0 0.0
bonds 2
harm  1 2 10.0 0.0
harm  2 3 10.0 0.0
finish
BC
nummols 3
beads 2
B  0.0 0.0 0.0
C  0.5 0.0 0.0
bonds 1
harm  1 2 10.0 0.0
finish

INTERACTIONS 3
A  A dpd   25.0 1.0 4.0
B  B dpd   25.0 1.0 4.0
C  C dpd   25.0 1.0 4.0

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

and no duplication is there (`nfold`= 1), the expected labeling of the beads [3] is

(correct order)

with the unbonded beads followed by the molecular ones. The latter are grouped into molecules as indicated. Please notice that the ordering of the beads within molecules *has to agree* with that given within the molecule definition, not with the ordering of the species (say, in the example above, A-C-B, not A-B-C!).

A possible CONFIG file for this example would look like this

```
-----------------------------------------------------
Simple test
 0       1
    1.0000000000    0.0000000000    0.0000000000
    0.0000000000    1.0000000000    0.0000000000
    0.0000000000    0.0000000000    1.0000000000
    B           1
    -0.35       0.46      0.05
    B           2
    0.27        -0.16     0.39
    B           3
    -0.04       -0.21     0.12
    ...
-----------------------------------------------------
```

Please notice that what is important is the label assigned to the bead, not the ordering of lines in the CONFIG files [4]. In other words, the previous CONFIG file can equivalently be written as:

```
-----------------------------------------------------
Simple test
 0       1
    1.0000000000    0.0000000000    0.0000000000
    0.0000000000    1.0000000000    0.0000000000
    0.0000000000    0.0000000000    1.0000000000
    B           2
    0.27        -0.16     0.39
    B           3
    -0.04       -0.21     0.12
    B           1
    -0.35       0.46      0.05
    ...
-----------------------------------------------------
```

with the bead number 1 described after beads number 2 and 3.

4 Volume definition

The system volume can either be defined in the CONTROL file (directive `vol`) or in the CONFIG file (using `imcon`> 0). A given configuration can be replicated along the Cartesian axes (`nfold` directive in the CONTROL file): please notice that this replication is incompatible with the use of the `vol` directive.

Case	Use `vol`	`nfold`	Use CONFIG	`imcon`
1	Yes	$= 1$	No	–
2	Yes	$= 1$	Yes	$= 0$
3	No	≥ 1	Yes	> 0

Table 1: Possible ways to define the system volume.

Also notice that if `imcon`> 0 three more lines (defining the system size) are expected in the CONFIG file, otherwise not. An inconsistency at this level would not be immediately detected, but would most probably result in some other error.

5 Absolute coordinates

Within DL_MESO_DPD, the particle coordinates are in $[0, dimx] \times [0, dimy] \times [0, dimz]$. The coordinates in the CONFIG file can be given in two different frames:

- $[0, dimx] \times [0, dimy] \times [0, dimz]$ a frame having origin in a corner of the system
- $[-dimx/2, dimx/2] \times [-dimy/2, dimy/2] \times [-dimz/2, dimz/2]$ a frame having the origin at the box center.

For the first case, the directive `conf zero` must be used in the CONTROL file.

What happens if the particle coordinates are out of the expected volume? They will be "wrapped" appropriately (i.e., corrected modulo *dimx*, *dimy*, *dimz*) by DL_MESO_DPD. However, the present utility will rise a warning each time a bead coordinate in the CONFIG file is out of the expected range.

6 Purpose of the module

The DL_MESO_DPD input files (CONTROL, FIELD and CONFIG) can be written by hand (doable for small systems) or generated using some tool (DL_MESO_DPD utilities and GUI, or GUIs from related projects, as DL_POLY [5] or Aten [6]). For hand-written files, it is useful to check for consistency between the files before running a simulation: this allows to detect certain mistakes which would not prevent DL_MESO_DPD from running, but would lead to wrong results.

Automatically generated input files will in principle be individually correct, but it is still a good idea to check that their combination does not lead to conflicts.

6.1 Bead labeling check

In its current version (2.6), DL_MESO_DPD checks that the total number of beads in the CONFIG file coincides with what expected from the FIELD file (possibly rescaled by `nfold` from the CONTROL file). If these differ, an error message is given and the code stops.

However, the code does not detect an erroneous preparation of the CONFIG file which has the right total number of beads, but wrong concentrations of the species, or wrong ordering of the beads. This would lead to unwanted effects. In the example above, a possible erroneous CONFIG file with the right composition but the wrong ordering, would lead to this arrangement into molecules

label:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
species:	A	A	A	A	B	B	B	B	B	B	B	B	B	B	B	B	B	C	C	C	C	C	C	C
	unbonded						*mol1*			*mol2*			*mol3*			*mol4*			*mol5*		*mol6*		*mol7*	

(wrong order!)

which is very different from the desired one.

The main purpose of the present module is to avoid such mistakes in the preparation of the CONFIG file.

6.2 Other checks

The purpose of the other checks is to verify that the volume definition is unique and no stretching bond crosses a hard wall, as this would be an obviously unphysical situation.

Finally, a warning is risen if the particle coordinates are not within the expected volume: this suggests to the user to confirm that the provided inputs for the system size and particle positions are indeed as intended.

7 Compiling

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

```
gfortran -o check_config.exe check_config.f90
```

and the executable must be in the same directory of the three files to be analyzed (i.e., CONTROL, FIELD and CONFIG).

8 Output

When running `check_config.f90`, the outcome of the different checks is sent to the standard output. The most important messages are: warnings, error messages and hints to fix them. For completeness, some information about the system size and composition is printed too.

In Table 2 we recall the meaning of the variables, for more details please see the DL_MESO package manual [7].

variable	type	description	values
`imcon`	integer	periodic boundary key for CONFIG file	$0, > 0$
`levcfg`	integer	data key for CONFIG file	$0, 1, 2$
`lconfzero`	logical	switch for using back bottom left corner as origin for CONFIG file	`T`, `F`
`srftype`	integer	surface type	$0, 1, 2, 3$
`nspe`	integer	number of species	≥ 1
`nmoldef`	integer	number of defined molecule types	
`mxmolsize`	integer	maximum number of particles per molecule	
`mxbonds`	integer	maximum number of bonds per molecule	
`nspec`	A, integer	species population of unbonded particles	
`nspecmol`	A, integer	species population of bonded particles	
`numbond`	integer	total number of bonds in system	
`mlstrtspe`	A, integer	species number for molecule insertion	

Table 2: Variables whose values are printed on the standard output by the utility.

9 Testing

We suggest as a test the same small system described above. In the first test, consistent input is given. In the following ones, small changes rising warnings and errors are analyzed, to demonstrate the behaviour of the module.

9.1 Test 1

Use for the CONTROL file

```
--------------------------------------------------
Simple test

temperature 1.0
cutoff 1.0

timestep 0.01
steps 1100
```

```
equilibration steps 100
trajectory 100 100 0
stats every 100
stack size 100
print every 100
job time 100.0
close time 10.0

ensemble nvt mdvv

nfold  1  1  1
#vol 1.0
conf zero
#surface hard z

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

and for the FIELD file the one given above in Section 3.

Using for the CONFIG file this (correct labeling) one, where the beads are randomly located in the cubic box.

```
-------------------------------------------------------
Simple test
       0       1
   1.0000000000    0.0000000000    0.0000000000
   0.0000000000    1.0000000000    0.0000000000
   0.0000000000    0.0000000000    1.0000000000
B    1
    0.4577200045    0.9001190080    0.3001750172
B    2
    0.0415166244    0.7699064654    0.8179705041
B    3
    0.6302680192    0.9146029274    0.7314079348
B    4
    0.7731659040    0.5993543351    0.3483148324
B    5
    0.1273913826    0.0669681234    0.5332509871
B    6
    0.3493437595    0.4205682036    0.4898004159
A    7
    0.0755944215    0.5154423406    0.0394230825
C    8
    0.5837477096    0.0477604149    0.7092934456
B    9
    0.0949452841    0.7453901460    0.7721903180
A    10
    0.2026802865    0.4475765512    0.4191000671
C    11
    0.8148312410    0.8686744347    0.8112311619
B    12
    0.3621449634    0.5704018599    0.7440643976
A    13
    0.4903370300    0.0944675650    0.7163648810
C    14
    0.9445609725    0.2362723351    0.0291370763
B    15
    0.7068423470    0.8993323711    0.0791676911
A    16
    0.5713842548    0.2551756180    0.7366135404
C    17
    0.2637800160    0.3507307479    0.7316829655
B    18
    0.3956074216    0.9739386044    0.9861309514
B    19
    0.9029085375    0.1837974484    0.0837168293
C    20
    0.0287508243    0.2151038377    0.2502012593
```

```
B    21
    0.1325665768    0.0464577116    0.8147593457
C    22
    0.6603299794    0.1659685862    0.4340299834
B    23
    0.7419336708    0.6792113832    0.5057230908
C    24
    0.6879917453    0.0772687141    0.6552782347
-------------------------------------------------------
```

Running the utility `check_config.f90`, this output is printed on the standard output:

```
-------------------------------------------------------
 unit cell sizes        =     1.0000000000     1.0000000000     1.0000000000
 nfoldx, nfoldy, nfoldz =        1        1        1
 system sizes           =     1.0000000000     1.0000000000     1.0000000000
 imcon     =        1
 levcfg    =        0
 lconfzero =        T
 srftype   =        0
 nspe      =        3
 nmoldef   =        2
 mxmolsize =        3
 mxbonds   =        2
 nspec     =        0        6        0
 nspecmol  =        4        7        7
 numbond   =       11
 for molecule ACB      :
 mlstrtspe =        1        3        2
 for molecule BC       :
 mlstrtspe =        2        3        0

 OK: CONFIG file is consistent with FIELD file
 (composition and bead content of molecules)
-------------------------------------------------------
```

9.2 Test 2

Instead, altering just two particle species in the CONFIG file given above:

- "B 3" into "A 3"
- "C 20" into "B 20"

an error message is given:

```
-------------------------------------------------------------------------------------------------------------------------------------------------------
 error: problem with unbonded beads of species A       :           1  instead of            0
 error: problem with unbonded beads of species B       :           5  instead of            6
 error: problem with molecular beads of species B       :           8  instead of            7
 error: problem with molecular beads of species C       :           6  instead of            7
 error: problem with the molecular content of  BC      :           2 -th bead is B        instead of C       (bead label =          20 )

 error: CONFIG file is not consistent with FIELD file
-------------------------------------------------------------------------------------------------------------------------------------------------------
```

9.3 Test 3

If instead these two lines of the CONFIG file are altered:

- "A 10" into "C 10"
- "C 11" into "A 11"

the error message is

```
-------------------------------------------------------------------------------------------------------------------------------------------------------
 error: problem with the molecular content of  ACB     :           1 -th bead is C        instead of A       (bead label =          10 )
 error: problem with the molecular content of  ACB     :           2 -th bead is A        instead of C       (bead label =          11 )

 error: CONFIG file is not consistent with FIELD file
-------------------------------------------------------------------------------------------------------------------------------------------------------
```

9.4 Test 4

Here instead we propose to add a hard wall orthogonal to the z axis: this is done uncommenting the "surface hard z" line in the CONTROL file. Running the utility, one obtains:

```
-------------------------------------------------------------------------------------------------------------------------------------------------------
...
 srftype   =        1
 srfx, srfy, srfz =           0           0           1
...
error: bond between beads        7  and        8  crosses hard wall perp. to z
error: bond between beads       13  and       14  crosses hard wall perp. to z

error: at least one streching bond is crossing a hard wall, please correct the CONFIG file
-------------------------------------------------------------------------------------------------------------------------------------------------------
```

Indeed it can be easily verified that for these two pairs of beads the bonds in the z direction have a length larger than half box: in this case (due to the minimum image convention), the bond will be built between a particle in the simulated volume and a particle in the adjacent cell, so crossing the hard wall.

10 Module building blocks

This module was built combining simplified versions of the subroutines which deal with the input files in DL_MESO_DPD (i.e., `read_control`, `scan_field`, `read_field`, `scan_config`, `read_config`), which are all contained in the modules `start_module.f90` and `config_module.f90`, and parsing utilities from the modules `parse_utils.f90`. Various checks were then added on top of this input material.

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).

[3] As expected by the DL_MESO_DPD subroutine building the molecules (which, among other things, is also assigning bonds). This subroutine is `read_config`, contained in the module `start_module.f90`.

[4] If no label is provided, one will be assigned based on the ordering. Also, in DL_MESO_DPD it is possible to use a directive (`no index`) to ignore particle labels in the CONFIG file.

[5] `https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx`

[6] `https://www.projectaten.com/`

[7] M. A. Seaton and W. Smith, *DL_MESO User Manual* - Version 2.6, November 2015.