OCTP 2.0 guidelines and proposal

Vladimir Jelle Lagerweij*

November 18, 2022

This document holds the derivations for the discretized equations of the Deoxer. First the overall continuous differential equation is derived, after which 3 different discretization schemes are proposed.
Jelle Lagerweij

1 Nomenclature

For this assessment of a deoxer, the following characteristics are used:

2 Goals for the input parameters

currently, the ordern diffusion has to have the following input:

```
compute c_ID all position  # computing the positions of all atoms and send them to master cpu.
fix ID group-ID ordern diffusivity Nevery Nfreq c_ID keyword values ...  # executes ordern algorithm
```

Listing 1: The basic lammps command parameters of current fix_ordern code with diffusivity mode selected.

For the diffusion coefficient, the value parameter has to refer to a compute position compute (part of OCTP plug-in code). However this should be replaced by a chunk compute (already part of basic Lammps code). Additionally, there should be more than one `c_ID` input allowed. For diffusion it should get the following workflow.

```
compute c_ID1 group-ID1 chunk/atom molecule nchunk once ids once  # computing which atoms are in molecule group-ID 1
compute c_ID2 group-ID2 chunk/atom molecule nchunk once ids once  # computing which atoms are in molecule group-ID 2
...
compute c_IDn group-IDn chunk/atom molecule nchunk once ids once  # computing which atoms are in molecule group-ID n

fix ID group-ID ordern diffusivity Nevery Nfreq c_ID1 c_ID2 ... c_IDn keyword values ...  # executes ordern algorithm
```

Listing 2: Final input goal for the improved OCTP plug-in. n different chunk types shall be allowed as input, for which then the sell diffusivity and Onsager coefficients shall be computed. The `nchunk once ids once` makes sure that the chunks compute will only assign atoms to certain chunks once and not change their designation for consistency.

3 Algorithm goals

If every chunk type gets a list with only their centre of masses, the mask style implementation with groups of the current algorithm can be removed as well. This should then result in the following work flow.

Hand drawing for now –> first current work flow

Hand drawing for now –> then future work flow

*Corresponding author. e-mail: v.j.lagerweij@student.tudelft.nl

4 Important variables and how to fetch them

Name	How it is retrieved	Meaning	also in .h	Change needed	How improvement is retrieved
`mode`	input arg 3 `arg[3]`	If ordern computes viscosity, T-conductivity or diffusivity	yes	no	
`idcompute`	input arg 6 `arg[6]`	the compute for which the ordern algorithm asks.	no	yes	should be multiple idcomputes possible: ¿¿¿how???
`icompute`	`=modify->find_compute(idcompute)`	real (lammps understanding) id of compute function	yes	yes	should be multiple icompute possible
`nrows`	`=modify->compute[icompute]->size_vector`	unclear yet (size of data passed from compute?)	yes	?	?
`count`	self made variable	number of samples taken, it just adds 1	yes	no	
`tngroup`	`=group->ngroup`	total number of groups	yes	yes	Shall be retrieved from number of computes in input. (=number of chunk types used.)
`ngroup`	self made variable	increases per indexed group	yes	yes	Can probably be removed if group numbering comes from chunk type.
`tnatom`	`=atom_>natoms`	total number of atoms	yes	yes	should be total number of chunks (full total, or total per chunk type). If `cchunk` refers to an existing chunk type, then `cchunk->setup_chunk()` might work.
`*cvector`	`double *cvector=ccompute->vector`	vector out compute (unclear to me why the `double` is needed)	no	yes	We (again) need to be able to read multiple chunk types

Table 1: The current names of the variables and how they can be retrieved are displayed in this table. Additionally, comments are made for when they need to be changed for OCTP 2.0.

5 Questions

1. Why is the for loops in the real ordern calculation `cor( k=1; k<=ngroup; k++)` with `ngroup` instead of the total amount of groups: `tngroup`?