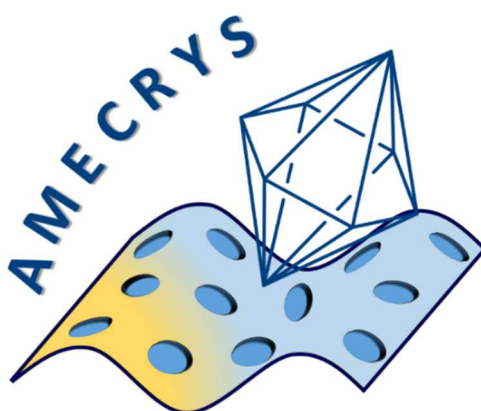


Deliverable number	D5.1	Due date	30/09/2017
Deliverable title	Simulation code for thermodynamics of course-grained model of mAbs in confined geometry	Issue date	29/09/2017
WP number	WP5	Author(s)	J. F. Lutsko, J. Lam
Lead Beneficiary	ULB	Reviewer(s)	All Partners
Deliverable type	Other	Status	Submitted
Dissemination level	Public		

AMECRYS - Revolutionising Downstream Processing of Monoclonal Antibodies by Continuous Template-Assisted Membrane Crystallization



Deliverables D5.1

Simulation code for thermodynamics of course-grained model of mAbs in confined geometry

Version	Modifications	Date	Author(s)
1.0	Initial document creation	01/09/2017	J. F. Lutsko
2.0	First draft	08/09/2017	J. F. Lutsko
3.0	Final version	22/09/2017	J. F. Lutsko, J. Lam



Licensed under Creative Commons Attribution - Non Commercial - No Derivatives 4.0 International
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



Contents

<i>Introduction</i>	3
<i>Description of code</i>	3
<i>Installation</i>	3
<i>Applications</i>	4
Liquid/Vapor surface tension.....	4
Liquid droplet equilibration on a solid substrate	5
<i>Limitations and future developments</i>	6
<i>Appendix: Phase diagram for entry of a nanoparticle into a pore</i>	6



Introduction

This deliverable consists of the finite temperature Density Functional Theory code written by the ULB team in the context of the AMECRYS project and the phase diagrams determined using it. This initial draft is being prepared at month 12 of the project and the relevant task, 5.1, continues until month 24 so the development will continue.

Description of code

The ftDFT code is a library that performs ftDFT calculations in a fully three dimensional, periodic geometry. The code is able to describe vapor, liquid and solid (crystal and amorphous) phases simultaneously. Examples included in the deliverable are (1) the equilibrium structure and free energy of a liquid droplet on a wall and (2) the equilibrium structure and excess free energy (surface tension) of a liquid-vapor interface. The current results obtained using the code to study the forces acting on a nanoparticle near a pore are detailed in a draft publication, which is included in the deliverable (Nanoparticle_phase_diagrams.pdf, found in the Documentation directory).

The general DFT framework (Fundamental Measure Theory with mean field corrections) is described in "James F. Lutsko. Recent Developments in Classical Density Functional Theory. Adv. Chem. Phys., 144:1, 2010."

3

Installation

Download the file AMECRYS_DELIVERABLE_5.1_code_v0.1.tar.gz (from AMECRYS Project website: <http://www.amecrys-project.eu/index.php/dissemination>) and run the command

```
tar zxvf AMECRYS_DELIVERABLE_5.1_code_v0.1.tar.gz
```

to create the root directory AMECRYS_DELIVERABLE_5.1_code in which the code, documentation and examples can be found.

The code was compiled under Ubuntu 16.04 and requires that the following libraries be installed prior to compilation:

- MathGL(mgl)
- FLTK
- Gnu Scientific Library (GSL)

- FFTW
- xmGrace
- Atlas BLAS Library
- Armadillo linear algebra library

Applications

In general, the ftDFT code can output the free energy and the density profile for a specific geometry (flat wall, cylinder, pore...) at a given temperature and chemical potential. In this paragraph, we will describe how the code can be used with two examples.

Liquid/Vapor surface tension

The presence of a liquid/vapor interface produces the emergence of a surface energy denoted γ that can be calculated within the classical DFT framework. In practice, DFT calculations are carried out within a volume V which is divided in two volumes loaded with gas and liquid thus forming an interface of surface S . The total free energy is then given by:

$$\Omega_{\text{tot}} = \omega_l V/2 + \omega_g V/2 + \gamma S$$

Where ω_l and ω_g are the bulk free energy per unit of volume of respectively the liquid and the gas.

In a new folder, copy the content of LiquidVapor. Three files are important to read:

- input.dat: Some parameters are read from this file. In particular, one has to be careful with those ones:
 - kT: Temperature in Lennard-Jones units
 - PointsPerHardSphere: Gridding discretization
 - Lx, Ly, Lz: Simulation box in three dimensions in Lennard-Jones units
 - rcut: Cut-off distance for the Lennard-Jones potential in Lennard-Jones units
- Periodic.h: This header file initializes the Density object. In practice, for the liquid/vapor application, the initialize function settles the initial density within the calculation box and here, one half of the simulation box is filled with liquid.
- liquid_vapor.cpp: For this application, one has to focus on few variables that are computed by the code:
 - xliq_coex, xgas_coex are respectively the liquid and gas density at coexistence
 - mu is the chemical potential at coexistence

- ω_{coex} is the bulk free energy per unit of volume
- `cg.getF()` outputs the value of the Free energy

The code is compiled by typing `make` and run using `./liquid_vapor input.dat` Here is the typical density profile that can be obtained:

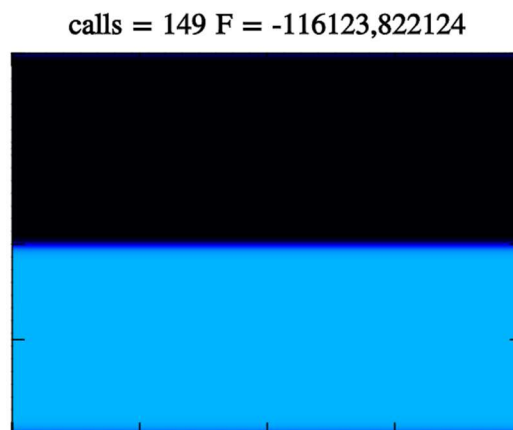


Figure 1 Resulting density in a typical liquid-vapor system. Blue is a liquid density and black indicates a vapor: the interface between the two has a width of about three atomic diameters.

Liquid droplet equilibration on a solid substrate

Droplet density profiles can be computed within the classical DFT framework. In practice, DFT calculations are started with a liquid spherical cap surrounded by gas disposed onto a solid modelled by Steele potential interactions. The strength of the wall/liquid interaction drives the wall hydrophobicity. Calculations are carried out at fixed number of particle to maintain the droplet stability.

In a new folder copy the content of Droplet. Three files are important to read:

- `input_droplet.dat`: Some parameters are read from this file. In particular, one has to be careful with those ones:
 - `epsWall`: the energy parameter for Steele potential
 - `R`: droplet initial size
- `Droplet.h`: This header file initializes the Density object. In practice, for the droplet application, the constructor settles the external potential which is the Steele potential for this case. The initialize function gives the initial density within the calculation box and here, a spherical cap is filled with liquid.
- `droplet.cpp`: For this application, one has to focus on few points:

- NN is the number of particles which is kept fixed.
- `theDensity.writeDensity(dump)` writes the density in a `dump.dat`. But, one can also write an additional function to write only a 2D version of the density that can be processed more easily.

The code is compiled by typing `make` and run using `./droplet input_droplet.dat`



Figure 2 Density distribution of a liquid droplet on a smooth wall.

Limitations and future developments

The code is currently written for pure (single species) systems. Further development will extend this to multicomponent systems so as to be able to study crystallization in solution (including in porous geometries such as membranes) and to be able to run on GPUs.

Appendix: Phase diagram for entry of a nanoparticle into a pore

The initial application of the code has been to study the forces acting on a nanoparticle in solution near a pore. Nanoscopic pores are used in various systems to attract nanoparticles. In general, the behaviour is a result of two types of interactions: the material specific affinity and the solvent-mediated influence also called the depletion force. The latter is more universal but also much more complex to understand since it requires modelling both the nanoparticle and the solvent. Here, we employed classical density functional theory to determine the forces acting on a nanoparticle near a nanoscopic pore as a function of its hydrophobicity and its size. A simple capillary model is constructed to predict those depletion forces for various surface properties. For a nanoscopic pore, complexity arises from both the specific geometry and the fact that hydrophobic pores are not necessarily filled with liquid. Taking these effects into account and including electrostatic effects, we



establish a phase diagram describing the entrance and the rejection of the nanoparticle from the pore. The knowledge gained from these studies will be used to construct coarse-grained models of nanoparticle-pore interactions that will be used to study the phase behaviour of the nanoparticles.

