CATCH-U-DNA

Capturing non-Amplified Tumor Circulating DNA with Ultrasound Hydrodynamics

Project No: 737212

Deliverable

Deliverable Title: Hydrodynamic code resolving coarse-grained molecular description of liposomes in oscillatory flow

Partner in charge: UAM

Due date:	December 2019
Date of submission:	January 2020
Author:	R. Delgado-Buscalioni
Document history:	N/A

Table of Contents

Horizon 2020
European Union funding
for Research & Innovation

1. Introduction

The following pages present an overview of the numerical system developed for simulations of soft-matter systems in contact with the Quartz Crystal Microbalance (QCM) with an adapted version of Florencio Balboa Usabiaga's original FLUAM code [Balboa2012], which may be found at (https://github.com/fbusabiaga/fluam). Numerical calculations were carried out in the context of the European Union Funding for Research & Innovation project Catch-U-DNA: Capturing non-Amplified Tumor Circulating DNA with Ultrasound Hydrodynamics.

2. Objectives

The aim of the Catch-U-DNA project is the creation of a highly-sensitive acoustic sensor platform (based on the QCM) to detect circulating-tumor DNA for cancer diagnosis, prognosis and treatment. In contrast to current techniques, it will not rely on labour-intensive steps, as in PCR-based analyses.

Because the new technique exploits the hydrodynamic properties of DNA strands, it was necessary to examine the behaviour of immersed DNA strands tethered to a QCM surface in different configurations and determine their effects on the hydrodynamic drag exerted on the QCM. The sheer scale of the systems involved (of the order of cubic microns, in order to contain at least one DNA strand) and the relevant time-scales (microseconds, for several oscillations of the QCM) precluded the possibility of using standard molecular dynamics codes to describe the time evolution of all the atoms involved. Although we did carry out some work on the behaviour of protein linkers adsorbed on the QCM surface, for most of our work we needed a coarse-grained approach capable of coupling the nanoscopic biological systems to the surrounding water. Hence, we adapted our group's FLUAM code for fluctuating hydrodynamics and immersed matter, based on Peskin's Immersed Boundary Method [Peskin2002], to the specific needs of this project.

3. Theory

In simple terms, the QCM technique consists in bringing a sample into contact with an AT-cut quartz crystal probe. Through the inverse piezoelectric effect, electric fields cause shear waves on the crystal surface. If a thin film of matter covers the surface, the frequency of vibration decreases in proportion to the mass adsorbed, as proved originally by Sauerbrey [Sauerbrey1959]. When brought into contact with fluid systems [Nomura1982], in addition to experiencing a shift in the frequency of vibration, the QCM dissipates energy. Typically, both the dissipation (ΔD) and the

frequency shift (Δf) may be measured with a QCM setup and, from these signals, the acoustic ratio ($\Delta D/\Delta f$) may be calculated. For many types of suspended molecules, the value of the acoustic ratio is independent of the concentration of adsorbed molecules.

Even though experiments were carried out with QCM-D devices, in which the quartz surface performs ring-down oscillations, allowing for the measurement of dissipation and frequency shift due to the load, the simulations carry out forced vibrations instead. Preliminary ring-down simulations proved too costly in terms of computational time and did not generate data with sufficient precision for the purposes of our analyses. Nevertheless, the small load approximation provides a link between numerical calculations of the load impedance and experimental data [Johannsmann2008],

$$\Delta \hat{f} = \frac{i f_0}{\pi} \frac{Z_L}{Z_Q},$$

where Z_L stands for the load impedance in the simulated system, the fundamental frequency equals $f_0 = 5$ MHz, and the quartz crystal impedance equals $Z_Q = 8.8 \times 10^6$ kg/(m²s). The complex frequency shift $\Delta \hat{f} = \Delta f + i \frac{f}{2} \Delta D$ includes the information of both the frequency shift and the dissipation.

4. Simulations

To begin, we need to construct a numerical representation of the system that we wish to simulate. A solid horizontal surface represents the QCM (in another section, below, we explain two different methods to represent this surface). Fluid water fills the space above the QCM. A stationary fixed wall above the QCM marks the top boundary of the simulation box. We assume the system is periodic in the other two directions of space. Furthermore, we wish to simulate matter suspended in the fluid (liposomes and DNA strands) which might be attached to the surface of the QCM.

FLUAM incorporates an Eulerian solver for the motion of a fluid with a given density and viscosity [Balboa2012]. It identifies the values of hydrodynamic variables on a staggered grid and determines the motion of the fluid in response to different initial conditions and forces by different methods. In the simulations described here, the relevant methods are named *particlesWall* and *quasiNeutrallyBuoyant*. The latter solves the discretised equations of motion for an incompressible fluid, while the former allows for the propagation of sound waves.

To describe matter immersed in the fluid, FLUAM uses the Immersed Boundary Method [Peskin2002], which consists in using a kernel to spread the forces on the

particles (imagined as points or small spheres) to the fluid. The solver then calculates the updated velocities with which the fluid responds to the forces exerted. To determine the effect on the masses, the code interpolates the velocities from the grid to the position of the particles and moves them accordingly.

Complex structures are built by connecting many immersed particles by means of linear springs (elastic network model). Liposomes are made out of the particles in a face-centred cubic lattice that lie between two imaginary spheres. The particles at these selected positions are then attached by means of harmonic springs. Polymer bead chains represent DNA strands. An extra angular potential defined for every three consecutive beads, chosen appropriately, reproduces the right persistence length for the chain. The DNA was fixed to the QCM plane with a linear spring. When the liposomes were tethered to the plane by means of DNA strands, extra springs connected each liposome to a strand.

In the incompressible scheme (*quasiNeutrallyBuoyant*) the walls were made up of two layers of a hexagonal close-packed lattice connected by linear springs. Furthermore, a vertical spring-like potential held them in place and they also had an additional harmonic potential attaching each of them to their initial position.

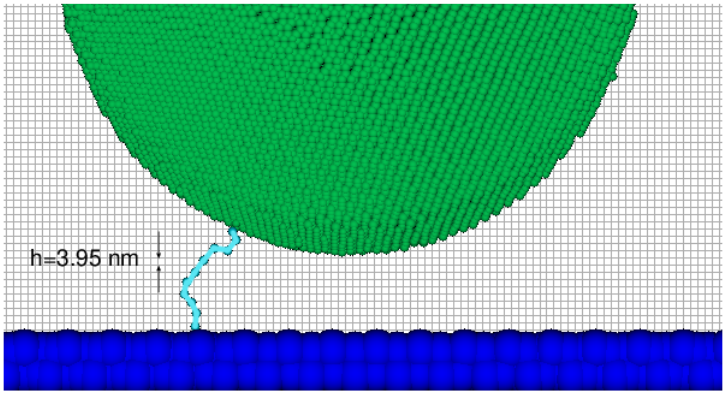

Figure 1: Example of an initial configuration for a FLUAM simulation of a liposome (green) tethered by DNA (light blue) to the Quartz Crystal Microbalance (dark blue). The grid shows the size of the fluid cells used by the Eulerian integrator for fluctuating hydrodynamics.

FLUAM simulations of the system in contact with the QCM could be setup and carried out by means of the **setup_simulations.sh** script, the parameters needed to run it will be explained below, but first we will explain the components on which it is based.

Simulation protocol

Numerical analysis of a system representing a sample composed of water with immersed structures (typically suspended liposomes and tethered DNA strands)

follows a three-step process. First, a numerical representation of the system in the initial configuration is created. Next, the data is fed into FLUAM, which works out the time evolution of the system in response to the vibrations of the QCM for several oscillation periods. Finally, post-processing scripts analyse the data output by FLUAM to determine the value of the load impedance.

Generating the initial conditions

When the QCM begins to vibrate in an experiment, the configurations of the liposomes and DNA strands are presumably representative of equilibrium configurations. Therefore, each simulation should begin with a representative configuration. Our initialisation scripts generate many configurations typical of equilibrium, which may then be used for simulations. An interesting conclusion of our research is that we can work out a good first approximation of the QCM impedance of a system of tethered liposomes by first calculating the equilibrium distribution of liposome heights and then using it as a weight factor to average over the values of impedance as a function of height.

Strand and liposome configuration

In our preliminary research, we used very long Brownian Dynamics simulations to determine the initial configurations. Later, we worked out a simpler method using Monte Carlo sampling. In essence, we generate a Brownian trajectory of the strand with the right persistence length. This trajectory represents its configuration. At one end, we choose a random orientation and place the liposome there. If two beads in the strand overlap, if they cross the QCM plane or if the liposome overlaps the chain or the QCM, we discard the configuration and start again. The program **liposome_configuration.c** generates these initial states. When executed without any parameters it prints out the following message:

```
Usage: ./liposome_configuration <number of configurations> <number of
monomers> <persistence length> <liposome radius> <linker energy>
<monomer colour> <liposome colour>

Distances are all measured in monomer diameters.
```

The parameters required by the programs are (in the following order) the number of configurations to generate, the number of monomers making up the DNA strand, the persistence length (measured in monomer diameters), the radius of the liposome (in monomer diameters), the bending energy at the point of contact between the strand and the wall, and the monomer and liposome “colours” which are just integers used to discriminate different types of particles.

The program prints out plain text data with columns specifying the *x*, *y* and *z* position of spheres, followed by the radius and type. Each configuration is followed by a blank line. For example,

```
> ./liposome_configuration 10 7 4 5 0 0 1
0.161783    0.615270    0.771537    0.500000    0
0.062528    1.073699    1.655630    0.500000    0
-0.240382   1.631214    2.429689    0.500000    0
-0.642972   1.756474    3.350563    0.500000    0
-0.906005   1.916005    4.306486    0.500000    0
-1.062933   2.398842    5.175434    0.500000    0
-1.532444   3.205856    5.555506    0.500000    0
-3.075348   3.877722    10.263801   5.000000    1

0.099016    -0.723106   0.683603    0.500000    0
                      ...
```

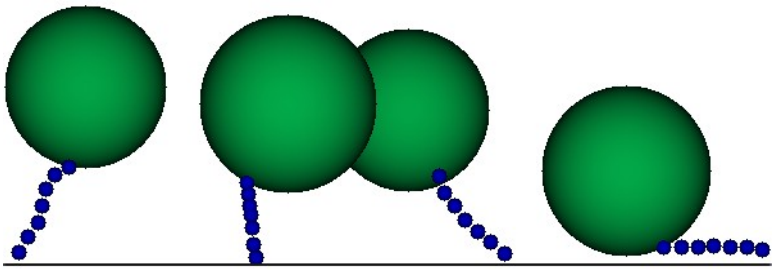

Figure 2: The first four random configurations of tethered liposomes generated with the liposome_configuration code (parameters set as in the example above).

A very similar code named **strand_configuration.c** generates polymer bead configurations with the right persistence length, but it only outputs a list of coordinates.

Replacing the liposome sphere with a multiblob structure

Within the numerical model, the liposome needs a representation with the right resolution, so we replace the tethered sphere generated by the previous code (see Fig. 2) by inserting a multiblob model of the liposome (as in Fig. 1). Solid three-dimensional structures were all generated with the **lattice.c** code, which allows us to create different types of lattice structures. When run without any arguments, it outputs a list of options.

```
> ./lattice
-- Bravais lattices --

Usage: ./lattice [options]

Options:
--type -t type               Lattice type (sc, bcc, fcc, dia, hcp,
sq or tri).
--number -N integer          Number of lattice nodes to generate.
--length -L number           Box side length.
-Lx -Ly -Lz number           Box length, width and height.
--radius -r number           Particle radius.
```

```
--colour --color -c integer   Particle colour (rgb integer).
--keep-aspect -k        Keep aspect ratio.
--fill -f number        Fill the box with particles separated by a
distance = <number>.
-b file                       User defined basis file.
-v file                       User defined lattice vectors file.
```

For the liposome, we create a face-centred cubic (fcc) structure within a cube and keep the points that lie between two spherical surfaces defined by the inner and outer radius of the liposome. A simple script called make_ball.sh carries out this job by invoking the Bravais lattice program with the appropriate parameters.

```
> ./make_ball.sh
Usage: ./make_ball.sh <inner radius> <outer radius> <separation> <x0>
<y0> <z0> <colour>
```

Execute the script providing the inner and outer radius of the liposome (in simulation units) the minimum separation between blobs, the initial coordinates of the liposome centre (`x0, y0, z0`) and an integer (`colour`) to identify the blobs that make it up.

Walls

In the directions parallel to the QCM surface, the simulation included periodic boundary conditions, but the perpendicular direction was different. For the compressible scheme (*particlesWall*), we relied on boundary conditions for the flow, fixing the velocity of the fluid at the lower and upper boundary of the simulation box. In contrast, the incompressible scheme (*quasiNeutrallyBuoyant*) introduced a multiblob model of the walls, with two layers of a hexagonal close-packed lattice for each wall generated by the code mentioned above. The bottom wall was then forced to vibrate during the FLUAM simulation run, while the top wall was held stationary at a distance large enough to guarantee that it had no significant effect on the measured load impedance.

An important difference between the schemes is that, while the compressible simulations assume that the walls are perpendicular to the *y* axis, the incompressible simulations operate with walls perpendicular to the *z* axis. Although trivial from the theoretical point of view, details like these often become the source of headaches in practice.

Elastic network models for walls, DNA strands and liposomes

Finally, the particles making up the walls, liposomes and DNA strands must be connected by bonds. FLUAM software accepts three types of bonds: fixed bonds, harmonic bonds and angular bonds. Fixed bonds are used to tether a particle to a fixed point in space. We use these bonds to hold the walls in place in the incompressible scheme simulations. Harmonic bonds are Hooke law potentials

between two given particles for which we specify the spring strength and the equilibrium length. Finally, angular springs connect three particles and oppose forces that attempt to change the angle they form. To define the latter bonds, we must specify the particles involved (in the right order), the angular bending energy and the equilibrium angle.

To create fixed bonds for the walls, we simply attach each wall particle to its initial position with the desired spring strength and a null equilibrium length.

Harmonic bonds were worked out with the **elastic_network.c** code, which creates a list of bonds attaching any pair of particles within a cutoff range specified by the parameter `Rc`.

```
Usage: ./elastic_network <N> <Rc> <K> <Lx> <Ly> <Lz> <file> [DIM]
Parameters:
      N:    Number of particles to read from file.
      Rc:   Cut-off radius for bonds.
      K:    Bond strength parameter.
      Lx, Ly, Lz: Box dimensions (enter -1 for no periodic boundary
conditions).
      file: Filename of particle positions (Format: x y z ... by
rows).
      DIM:  Dimensionality of space (1, 2 or 3).
```

The code outputs a list of bonds with one bond on each line. The first two columns indicate the particle indices, followed by the spring strength and equilibrium length.

Finally, angular springs were created for the DNA by attaching every three consecutive beads in the strand and choosing the bending energy in such a way that the strands had the right persistence length.

For tethered structures, additional harmonic bonds must connect the DNA strand to the liposome and QCM surface.

Numerical forced QCM vibrations

The adapted FLUAM code simulates forced vibrations of the QCM surface and calculates the effect on the water and liposomes by integrating the equations of motion for the fluid and suspended structures. In order to run a FLUAM simulation, we need four files: one (called data.main) contains the simulation parameters, the second holds the initial configuration of the system, the third contains a list of the harmonic and fixed bonds, and the final file includes the information for the angular bonds. We will turn to the configuration and bond files first.

Initial configuration file

The positions of all the particles at time $t=0$ should be specified in a plain text file. The following example shows the format of the initial configuration file.

```
> head initial_configuration.dat
5576
-16.000000 -16.000000 63.75 250
-15.500000 -15.750335 64.25 250
-15.500000 -15.135135 63.75 250
-15.000000 -14.885470 64.25 250
-15.000000 -14.270270 63.75 250
                ...
```

The first row includes a single integer indicating the number of particles listed in the file. The following lines list the coordinates and colour (or type) of each particle. Floating point values indicate the x, y and z coordinates of each point in simulation units. The final column contains an integer for the colour. Note that FLUAM files do not have a column to specify the radius of each particle, as the Immersed Boundary Method algorithm enforces a hydrodynamic radius for each particle of the order of the fluid cell width. In addition to the hydrodynamic radius, an additional radius for Lennard-Jones-type excluded volume interactions may be set independently for pairs of particle types (that is, type 0 with type 1, or type 2 with itself).

Bond files

The format for FLUAM fixed and harmonic bond files follows a similar scheme. First, an integer indicates how many harmonic bond entries are listed, followed by the list of bonds. Then, another integer specifies how many fixed bonds to read, followed by the list of fixed bonds.

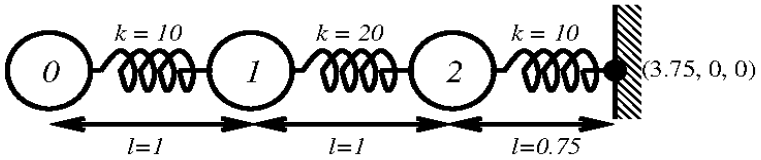

Figure 3: Simple three-particle configuration with two harmonic bonds and one fixed bond connecting particle 2 to point (3.75, 0, 0).

For example, the bond file for the arrangement seen in Fig. 3 would look like this:

```
4
0 1 10.0 1.0
1 0 10.0 1.0
1 2 20.0 1.0
2 1 20.0 1.0
1
2 10 0.75 3.75 0 0
```

Several important warnings come to mind here. First, note that every harmonic bond appears twice (once for each particle involved) and that means that the integer at the top of the file should equal twice the number of bonds. Second, harmonic bond entries are formatted into four columns (two integers and then two real numbers) separated by white space: *i j k l*, where *i* stands for the particle on which the bond acts, *j* the other particle in the bond, *k* the spring constant and *l* the equilibrium length. Third, particle indices start at 0 and their order corresponds to the order in which they appear listed in the initial configuration file. Fourth, bond entries are ordered by first and second columns (FLUAM expects this order and will otherwise print out an error message). Finally, the fixed bond format is *i k l x y z*, with *i* the particle index, *k* the spring constant, *l* the equilibrium length and (*x*, *y*, *z*) the fixed point to which the particle remains tethered.

Parameter file

To execute FLUAM, provide a file named **data.main** in the same folder as the executable containing the simulation run parameters. The list of parameters depends on the integration scheme, although many parameters are shared between both schemes. We will begin with the compressible case. There are two versions of the program. In one version, we set the velocity of the QCM wall to follow $v(t)=v_0\cos(\omega_0 t)$, with v_0 representing the velocity amplitude and ω_0 the angular frequency. Furthermore, we impose stick boundary conditions on the water, so the simulation cells in contact with the wall will have velocities given by $v(t)$. In the second version, we apply a periodic force to the wall $F(t)=F_0\cos(\omega_0 t)$. Therefore, below we provide a complete **data.main** file for the compressible scheme (*particlesWall*) imposing the velocity of the QCM wall, followed by some comments on how the second compressible scheme differs from the former.

```
#############################################################
# FLUAM data.main file containing simulation parameters #
#############################################################

# Uncomment to select the GPU on which to run FLUAM
# (in general it's better not to use this option)
# setDevice 0

##### Immersed particle parameters #####

# Are there immersed particles in the simulation?
# (1 = yes, 0 = no)
particles   1

# Do particles have an integer property indicating their type?
# (1 = yes, 0 = no)
colors      1

# Excess mass
# (mass = 0 for neutrally buoyant particles)
mass      0

# Activate Lennard-Jones-type interactions?
```

```
# (1 = yes, 0 = no)
computeNonBondedForces 0

# Cutoff for the particle-particle interaction
# (not relevant in this case because these forces
# have been deactivated in the previous option)
cutoff                  1.0

# Maximum number of particles in the neighbour list
maxNumberPartInCellNonBonded    20
maxNumberPartInCell       20

### Initial configuration for particles ###
# Load particle positions from file?
# (1 = yes, 0 = no)
# If no file is given the particles start on a simple cubic lattice.
loadparticles   1

# Coordinates file
coordinates   initial_configuration.dat

# Optional initial velocities files
# velocities   initial_velocities.dat

### Bond files ###
# File containing harmonic and fixed bonds
bondedForces    bonds.fluam.dat

# File for angular bonds
threeBondedForces bonds3.fluam.dat

##### Fluid parameters #####
# Dimensions of the simulation box in simulation units
celldimension   64  64  64

# Number of fluid cells in the simulation box
# in the x, y and z directions
cells     128 128 128

# Fluid density
densfluid               1

# Shear viscosity
shearviscosity          0.226194671058465

# Bulk viscosity
bulkviscosity           0

# Thermal energy in units of k_B*T
# (temperature = 0.0 calculates the hydrodynamics
# without any thermal fluctuations)
temperature             0.0
```

```
# Equation of state parameters. The pressure equals
# p = a0 + a1*density + a2*density**2
#                       a0  a1   a2
pressureparameters      0  4.0  0

# Fluid initial conditions
# initfluid = 0 means that the initial velocity
# equals the background velocity and that the density
# equals density_0, while initfluid = 1 means that
# the initial velocity equals the background velocity
# plus random fluctuations and that the density is
# equal to density_0 (see fluam/src/initializeFluid.cpp)
initfluid 1

# Background velocity components vx, vy, vz.
# If not provided, it will be set to (0, 0, 0).
#                     vx   vy   vz
backgroundvelocity   0    0    0

# Uncomment to read in the initial fluid configuration from a file.
# If not provided, fluam uses the option given in initfluid.
# fluid initial_fluid_configuration.dat

##### QCM wall parameters #####
# Forced vibration angular frequency imposed on the QCM surface
freqWall 0.00314159265358979

# Components of the fluid velocity at the lower and upper walls.
# vxWall represents the amplitude of the oscillating wall velocity.
# v(t) = vxWall*cos(freqWall t).
#           Lower wall          Upper wall
vxWall      0.00314159265358979 0

##### Integration parameters #####
# Chose integration scheme
# compressible   --> particlesWall
# incompressible --> quasiNeutrallyBuoyant
particlesWall

# Seed for the random number generator. When not provided,
# FLUAM takes one from the computer clock.
# seed     1

# Number of relaxation steps during which FLUAM saves no data
numstepsRelaxation  0

# Number of simulation steps (not counting numstepsRelaxation)
numsteps               250000

# Save data every samplefreq steps
samplefreq  1200
```

```
# Save fluid cells data every savefreq steps
# savefreq should be an integer multiple of samplefreq
savefreq    1200

# Integration time step
dt    0.025

# Save data in VTK format to be visualized with the program VisIt.
# See file saveFluidVTK.cu and the web URL
# https://wci.llnl.gov/simulation/computer-codes/visit/
saveVTK   0
saveFluid 0

# Prefix for the output files. The directory where the data is saved
# should already exist before running FLUAM
outputname    output_data/QCMrun1
```

In the second variation of the compressible scheme, where we apply a force to the wall, we need to include the following additional parameters:

```
# Mass of the QCM wall
massWall 256.0

# Amplitude of the oscillating force on the QCM wall
FWall 10.0

# Spring constant of the harmonic restoring force of the wall
kWall 0.0
```

Furthermore, the parameter `vxWall` now changes its meaning to the initial velocity of the walls.

Turning to the incompressible scheme (*quasiNeutrallyBuoyant*) we find some important differences. The example below includes a complete (though brief) **data.main** file for this scheme. Many of the parameters have the same meaning as in the compressible simulations. As will become obvious from reading the example, some of the parameters from the compressible example have disappeared, but others have been added.

Because this scheme codes the QCM wall in a completely different fashion, the parameters used to deal with it in the **data.main** file change. Recall that the incompressible scheme represents the upper and lower walls by means of particles in an hcp crystal structure. To hold them in place, a vertical harmonic potential attracts them to a given height. Below, a harmonic potential with spring constant equal to 10.0 holds the first 2368 particles at a height of $z=64$ (simulation units), a second potential pulls particles with indices between 2368 and 4735 towards $z=0$. Remember that particle indices start at 0, so the first 2368 particles refers to particles with indices in the range 0–2367, both ends included.

Furthermore, the force imposed on the wall obeys the formula $F(t)=F_0\sin(2\pi f t)$, where F_0 stands for the force amplitude parameter and f for the frequency. An extra parameter indicates the duration of the applied force. If the simulation carries on longer than the

duration of the force, then FLUAM wall will simulate a ring-down relaxation after the forcing stops.

```
#########################################################
# FLUAM data.main file containing simulation parameters #
#########################################################
# setDevice 0

##### Immersed particle parameters #####
# Activate particles
particles   1

# Excess mass of particles
mass 0

# Interaction forces
computeNonBondedForces 0
cutoff                 2.5

### Initial configuration for particles ###
# Load particles from file
loadparticles   1

# Include particle types
colors    1

# Initial configuration file
coordinates   initial_configuration.dat

# Optional initial velocities files
# velocities   initial_velocities.dat

# Memory allocation limits
maxNumberPartInCellNonBonded    64
maxNumberPartInCell       64

### Bond files ###
# File containing harmonic and fixed bonds
bondedForces               bonds.fluam.dat

# File for angular bonds
threeBondedForces bonds3.fluam.dat

##### Fluid properties #####
initfluid 1
densfluid               1.0
shearviscosity          1.0
temperature             0.0
saveFluid 0
saveVTK 0

##### QCM wall parameters #####
#Horizontal wall params   Last particle    K spring    z coordinate
```

```
wall1Params                2368              10.0         64
wall2Params                2368              10.0         64
wall3Params                4736              10.0         0.0

# QCM wall oscillation parameters.
# F(t) = F0 sin(2*pi*f*t), for 0 < t < t_f.
#Stokes flow parameters Frequency (f)  Amplitude (F0)  Duration (t_f)
stokesFlowParams        0.000796       .2              10000000

#### Numerical integration ####
# Chose integration scheme
# compressible   --> particlesWall
# incompressible --> quasiNeutrallyBuoyant
quasiNeutrallyBuoyant

#Dimensions of the simulation box
celldimension  32 32 128

#Number of fluid cells in the directions x, y and z
cells   32 32 128

# Time steps
numstepsRelaxation  0
numsteps            12562814
samplefreq          3140
savefreq            3140
dt                  0.002

# Prefix for output files
outputname    output_data/QCMrun1
```

Output data

Every `samplefreq` timesteps, FLUAM sends information including the temperature and pressure to the standard output.

```
INCOMPRESSIBLE BOUNDARY  ;) 3140
MASS FLUID 131072
TEMPERATURE FLUID 0.000146084 0.000146084 inf inf  1
tx                5.54547e-05 5.54547e-05
ty                0.000204767 0.000204767
tz                0.000178028 0.000178028
TEMPERATURE PARTICLE 0.000401328 0.000401328 inf
PX -13.3762613455386
PY 151.377978643525
PZ 20.1066389099346
```

The compressible (*particlesWall*) scheme executable also writes out information on the QCM wall:

```
wall 6 0.193048 0.02915
force 6 -7.39901 0 9.29776 -1.93048 -0.0311422
Particles Wall  4000
```

The first (wall) line uses the format: `wall t x v`, where `t` stands for the time, `x` for the wall position and `v` for its velocity. The second (force) line includes the information of forces acting on the wall and is formatted in columns in the following order: `force t Fvisc Fpol Fext Fk Ftot`, where `t` stands for the time, `Fvisc` for the viscous drag of the fluid, `Fpol` for the traction of the polymer strand, `Fext` for the external forcing, `Fk` for the restoring harmonic spring and, finally, the sum of all these force terms in `Ftot`. The last line includes the simulation step number in the particlesWall scheme (in this example, 4000).

Additional data generated by FLUAM will typically be contained in files named with the prefix indicated by the `outputname` parameter in **data.main**, followed by a dot and a data type specifier (for example, **QCMrun1.particles**, **QCMrun1.data.main**, **QCMrun1.gradv**, etc.). The **<prefix>.data.main** file is just a copy of the **data.main** file used to set up a simulation and it can be used to check the parameters for a given run, as the **data.main** file will usually change between runs.

Below, we will comment on the files created by an incompressible run. The **<prefix>.particles** file contains a list of particle configurations. The first line indicates the number of particles and is followed by a series of configurations. Each configuration begins with a hash and the simulation time, followed by an ordered list of particle entries. Each of the latter includes the *x*, *y* and *z* position of the particle and an integer indicating the particle type.

```
> head QCMrun1.particles
#NUMBER PARTICLES 4972
#6.28
-15.9586159450817 -16.2217869372719 63.6757598875441 250
-15.5138662898658 -15.9523909263096 64.3087583554132 250
-15.5765989506598 -15.4243586797628 63.5572067681578 250
-15.1710423779576 -15.1627687382027 64.189934335774 250
                              ...
```

The **<prefix>.velocityParticles** and **<prefix>.velocityParticlesI** files contain information on the particle velocities, which we tend to disregard in our analyses, so we often link these files to **/dev/null** to save space. Fluid velocities may also be stored by activating the `saveFluid`, `saveFreq` and (optionally) `saveVTK` options in **data.main**.

The file **<prefix>.gradv** contains important information for the calculation of the QCM load impedance. The three columns represent simulation time step, the derivative of the fluid velocity with respect to *z* at the wall, and the velocity of the fluid in contact with the wall, in that order.

```
> head QCMrun1.gradv
3140 0.00241563880791512 -0.00384860625333991
6280 0.00711392506975435 -0.0124469501964481
9420 0.00459200705651164 -0.0121171843963312
12560 0.00353274501599548 -0.00878417556485663
15700 0.00386752791574959 -0.00833739459406569
18840 0.00351077624298315 -0.00779248847707166
                     ...
```

Finally, **<prefix>.cuda** contains information on the graphics processing unit (GPU) and **<prefix>.seed** the seed used for the pseudorandom number generator.

Impedance analysis scripts

Post-processing the data generated during the simulation run reveals the motion of liposomes, flow profiles and many other magnitudes, among which is the main magnitude for comparison with experiments: the complex load impedance of the sample, Z_L, defined as the ratio of the shear stress on the QCM wall caused by the fluid load, $\sigma(t)$, to the QCM velocity, $v(t)$[Johannsmann2015],

$$Z_L(\omega)=\frac{\sigma(t)}{v(t)}.$$

We use complex phasors to represent both the shear stress and the velocity. For example, $v(t)=v_0 e^{-i\omega t}$, where v_0 stands for the complex amplitude and we understand that the velocity of the wall corresponds to the real part of $v(t)$. We provide an example analysis script to calculate the load impedance (**analysis.sh**).

Measuring the load impedance imposing the velocity of the QCM wall

The most direct way to determine the load impedance in a forced oscillation of the simulated QCM surface consists in imposing the velocity of the lower wall and measuring the resulting shear stress caused by the load. Therefore, we take it that the wall moves with velocity $v(t)=v_0\cos(\omega t)$ with stick boundary conditions for the flow, and we measure the derivative of the fluid velocity with its distance to the QCM surface in order to calculate the observed stress as

$$\Re[\sigma(t)]=\eta\frac{\partial u(z,t)}{\partial z}.$$

The function $\Re[z]$ means real part of *z*. Once we have the (numerical) observed stress as a function of time, we fit it to a function of the form $\sigma_r\cos(\omega t)+\sigma_i\sin(\omega t)$, and thereby determine the real and imaginary part of the complex amplitude of the stress, $\sigma_0=\sigma_r+i\sigma_i$.

This method appears to be appropriate in the absence of suspended structures, or when not too many immersed particles lie near the QCM surface interfering with the flow pattern.

Measuring the load impedance exerting a predefined force

When we specify the force on the wall, instead of its velocity (as in the incompressible scheme), we need to determine both the velocity and the shear stress. To work out the velocity, we track the centre of mass of the wall and fit it to a function of the form $x_r\cos(\omega t)+i\,x_i\sin(\omega t)$, which of course lets us write the position phasor as $x(t)=(x_r+i\,x_i)e^{-i\omega t}$. Because the velocity is the derivative of $x(t)$, we see that the complex velocity amplitude must equal $v_0=\omega(x_i-i\,x_r)$.

The shear stress could be measured as explained in the previous section, but in this setup there also exists another possibility. Because, according to Newton's second law,

$$m\frac{d^2x}{dt^2} = -m\omega^2 x = F(t) + A\sigma(t) + F_{spring},$$

where *A* stands for the wall area and F_{spring} for harmonic spring forces, the shear stress equals the difference between the fluid inertia and the other mechanical forces acting on the wall (which can be readily determined), divided by the area *A*.

Finite size effects

The presence of a stationary upper wall has an effect on the numerical load impedance measured in the simulation. Although negligible as soon as the distance between the QCM wall and the upper wall exceeds about 3 times the penetration depth of the oscillating boundary layer, it may be determined analytically and used to introduce corrections in the numerical result. The **QCM_two_walls.sh** script calculates the expected impedance for a fluid (without suspended particles) as a function of the distance between the two walls (Figure 4).

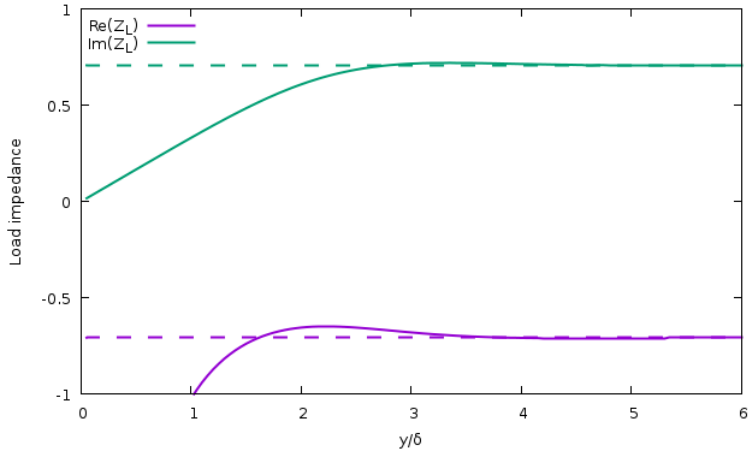

Figure 4: Load impedance of a fluid trapped between an oscillating QCM wall and a parallel stationary wall as a function of the distance, y, between walls. The dashed lines mark the values of the analytical solution for an unbounded domain.

Scripts to set up the simulations

The preceding pages should have made it obvious that the FLUAM simulations carried out for forced QCM vibrations depend on many crucial details, which might become the source of errors. For example, the bond file needs the entries ordered correctly and they must include the indices of the particles they bind together. Getting one of these indices wrong by one can potentially ruin the whole simulation.

Therefore, it is essential to connect the codes by means of scripts that guarantee that these details are handled correctly.

A typical simulation setup script for the QCM simulations should take care of the following general steps:

(1) Set the values of the list of parameters defining the desired simulation. It should allow for the possibility of sweeping over several values of one or more parameters.

(2) Calculate the values of other parameters which may be determined from the values given in step 1.

(3) Write the FLUAM **data.main** parameter file.

(4) Create the multiblob structures: DNA strands, liposomes and walls (if necessary).

(5) Create the elastic networks connecting blobs within structures and interconnecting structures (liposomes to DNA strands, and strands to walls).

(6) Work out representative initial configurations (equilibrium states). In order to improve statistics, a useful strategy involves launching many simulations in parallel with the same parameters but different initial configurations. The script should prepare each of these runs with all the necessary files in a separate directory.

(7) Finally, the script should launch the simulations.

Once FLUAM has finished the numerical work, analysis scripts extract the desired information from the output files. See **setup_simulations.sh** for an example setup script. The code begins with a list of parameters which define the simulation:

```
#!/bin/bash
#####################################
# FLUAM QCM simulation setup script #
#####################################

### Parameters ###
PARAM_SIMNAME="QCM_sim" # Simulation name
PARAM_NODE="0-9" # GPU node
PARAM_BDNODE="0-9" # CPU node
PARAM_CASE="run6" # Case name

# Experimental parameters
PARAM_LDNAEXP=50 # Length of DNA strand (nm)
PARAM_LPEREXP=50  # Persistence length of DNA strand (nm)
PARAM_DELTAEXP=95 # Penetration length of the oscillating fluid (nm)
PARAM_RADIUSEXP=50 # Liposome radius (nm)
```

```
PARAM_LIPOSOMETHICKNESSEXP=10 # Thickness of the liposome membrane
(nm)

# Constants
PARAM_DELTA=12 # Penetration length in simulation units (fixes the
simulation resolution)
PARAM_CELLSIZE=0.5 # Size of simulation cell in FLUAM.

# Files
PARAM_RUNSCRIPT="run.sh" # HPC Cluster run script for FLUAM
PARAM_PARAMETERFILE="data.main" # Fluam parameter file
PARAM_WALLFILE="wall.tmp" # Wall particle xy coordinates
# Initial configuration file name
PARAM_PARTICLEFILE="initial_configuration.dat"
PARAM_BONDFILE="bonds.fluam.dat" # Bond file name
PARAM_ANGULARSPRINGSFILE="bonds3.fluam.dat"

# Numerical integration
PARAM_TIMESTEP=0.002 # Integration time step
PARAM_NPERIODS=20 # Number of QCM oscillation periods
PARAM_NDATAPER=200 # Number of points per period to save

# System
PARAM_LX=32 # Simulation box length
PARAM_LY=32 # Simulation box width
PARAM_LZ=128 # Simulation box height

# Particles
PARAM_BOTTOMWALLC=255 # Colour for bottom wall
PARAM_TOPWALLC=250 # Colour for top wall
PARAM_LJEPSILON=0 # Lennard-Jones epsilon parameter
PARAM_EXCESSMASS=0 # Excess mass of particles

# Walls
PARAM_WALLBOND=10.0 # Wall bond parameter (holding the wall at
                    # the right z position --no xy force--).
PARAM_WALLAMP=1 # Amplitude of QCM oscillation
PARAM_WALLFORCETMAX=10000000 # Duration of forcing
PARAM_WALLFREQ=0.0025 # Frequency of forcing
PARAM_FBOND=0.2 # Fixed bond parameter for walls (holding each
                # wall particle to its initial position).
                # Set FBOND to 0 (integer zero) to ignore fixed bonds

# Polymers
PARAM_PBOND=100.0 # Polymer bond parameter
PARAM_CHAINC=123123 # 3    # Polymer chain colour

# Liposome
PARAM_NLIPOSOMES=0 # Number of liposomes to simulate
PARAM_LIPOSOMEC=321123 # 5 # Liposome colour
PARAM_LBOND=100.0 # Liposome bond parameter

# Fluid
```

```
PARAM_RHO=1 # Fluid density
PARAM_KT=1.0 # Thermal energy
PARAM_THERMALFLUCTUATIONS=N # Set to "N" to ignore
```

5. Results

This section includes a few representative examples of the kind of results provided by our framework for QCM research.

For starters, experimental research has established that low concentrations of nanospheres adsorbed on the QCM surface give rise to a much greater dissipation and frequency shift than one would expect from the Sauerbrey thin layer approach. For example, the impedance of adsorbed cowpea mosaic viruses (CPMV) was measured as a function of their concentration in [Johannsmann2009] (Fig. 5). Interestingly, in contrast to the behaviour observed for spheres suspended at a given height, higher harmonics lead to a greater dissipation.

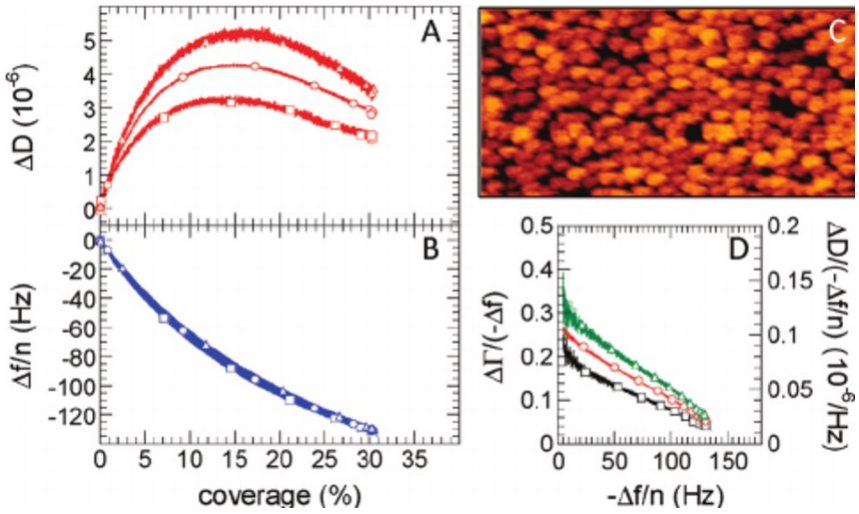

Figure 5: Dissipation (A) and frequency (B) measured by QCM-D as a function of the concentration of CPMV adsorbed onto a gold surface. Squares correspond to the fifth overtone of the fundamental frequency, circles to the ninth and triangles to the thirteenth. (C) Atomic force microscopy image of the surface after the measurement (1.0 x 0.5 square microns). (D) $\Delta\Gamma/(-\Delta f)$ *ratios as a function of frequency for the same adsorption process. Figure reproduced from [Johannsmann2009].*

QCM simulations with FLUAM display a very similar behaviour (as shown in Fig. 6). The magnitude and growing dissipation for increasing overtones appear clearly, although the simulations seem to overestimate the decay with concentration. This might, however, be due to a small amount of stacking of the spheres in the experiment, but this statement requires further investigation.

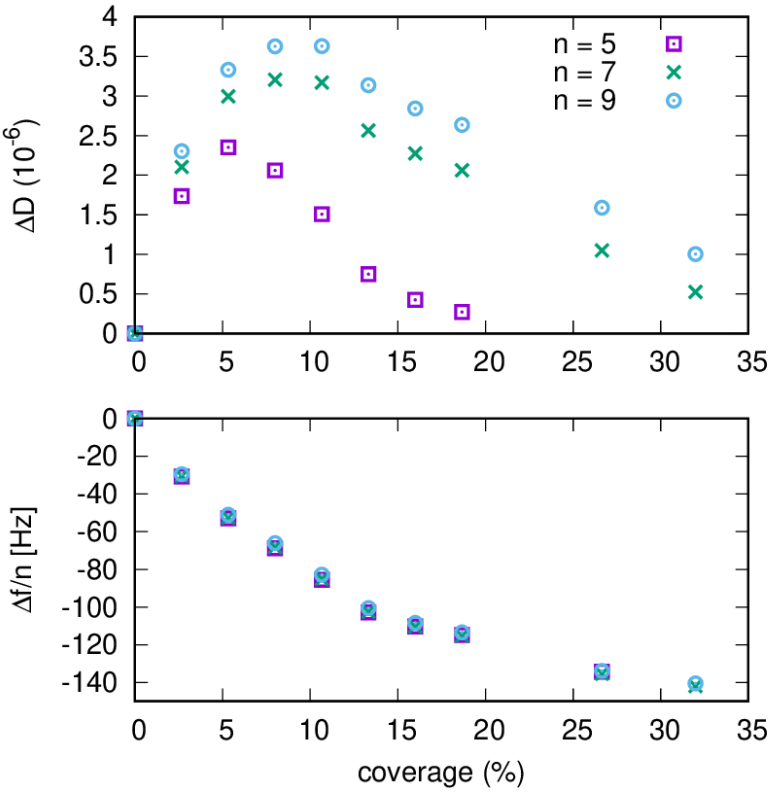

Figure 6: FLUAM simulation results for dissipation and frequency shifts of adsorbed CPMV on a QCM surface.

The simulations also make relevant contributions to theoretical research. In experiments, we expect liposomes tethered to the QCM surface at different heights. In the dilute regime, we have shown that in many cases it is possible to give a very good estimation of the load impedance and acoustic ratio by combining the information in the height distribution with the impedance as a function of distance [Delgado2020]. Analytically, we might wish to possess equations for the load impedance of the sphere as a function of its distance to the QCM. This information cannot be easily extracted from experiments, but may be readily determined from simulations. Fig. 7 plots load impedances for small spheres drawn from FLUAM QCM simulations as a function of their distance to the vibrating wall. The lines correspond to the analytical predictions of a very simple model for mass suspended at a distance *d* from the wall. Apart from a prefactor (points correspond to the right vertical axis, while lines correspond to the values on the left), the analytical theory perfectly mirrors the behaviour of the simulated liposomes.

As a final example of the detailed nature of the information made available by hydrodynamic simulations, we include in Figures 8 and 9 a plot of the perturbation of the fluid density and the fluid velocities around a liposome near the vibrating crystal with respect to the flow without any liposomes. Our investigation so far suggests that the effect of suspended matter on the hydrodynamic drag felt by the QCM constitutes the main contribution to the load impedance of tethered liposomes. The differences in density generate pressure gradients that affect the flow velocities seen in Fig. 9.

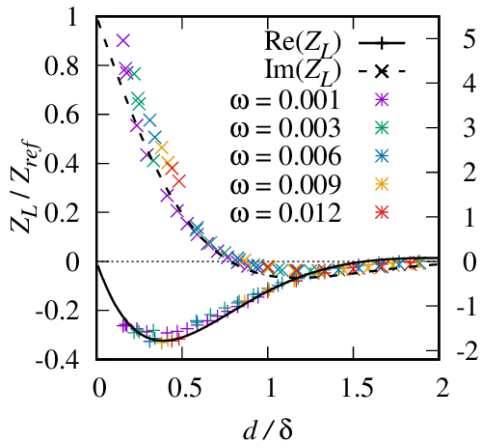

Figure 7: The real and imaginary part of the load impedance as a function of the distance to the QCM surface. Lines represent values on the left axis for the load impedance of an infinite plane. Points and the right axis show the values for small spheres ($R = 0.16\,\delta$) in a (1.33 x 1.33 x 5.34)δ^3 simulation box with periodic boundary conditions along the QCM plane. The different colours correspond to simulations carried out at different frequencies. Z_{ref} is a mass (of the plane or spheres) per unit surface of QCM multiplied by the angular frequency.

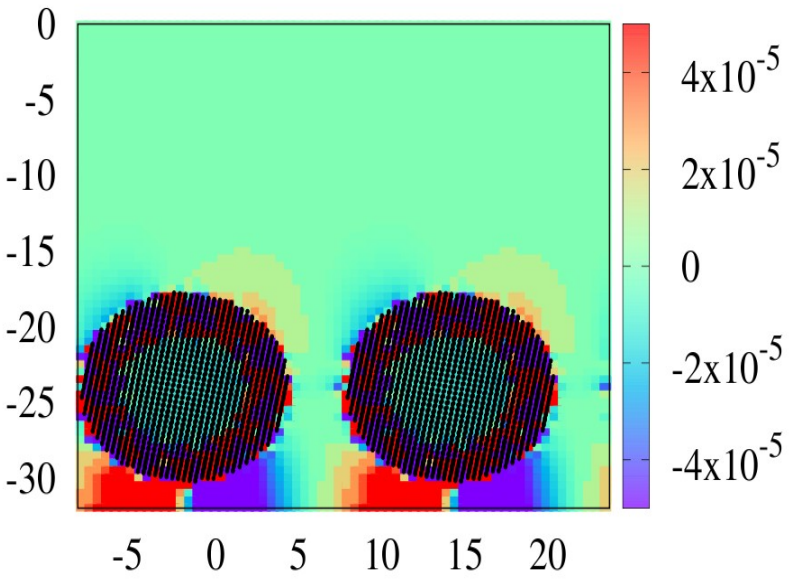

Figure 8: Perturbation of the density around two liposomes near a QCM surface with respect to the Stokes boundary layer flow without liposomes, at a 0 rad phase angle. The simulation corresponds to a flow with penetration length $\delta = 12$ in simulation units with a resolution of $h = 0.5$, which corresponds to 50-nm-radius liposomes in the flow of a QCM vibrating at 35 MHz.

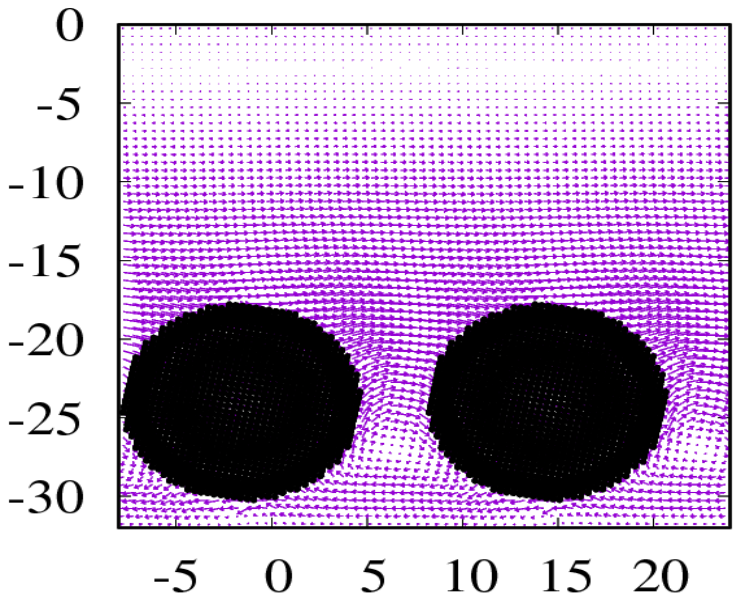

Figure 9: Perturbation of the flow velocity around two liposomes near a QCM surface with respect to the Stokes boundary layer flow without liposomes, at a 0 rad phase angle. The simulation corresponds to a flow with penetration length $\delta=12$ *in simulation units with a resolution of* $h=0.5$ *, which corresponds to 50-nm-radius liposomes in the flow of a QCM vibrating at 35 MHz.*

6. References

[Balboa2012] F. Balboa Usabiaga, J. B. Bell, R. Delgado-Buscalioni, A. Donev, T. G. Fai, B. E. Griffith, and C. S. Peskin: *Staggered Schemes for Fluctuating Hydrodynamics*, *Multiscale Modeling & Simulation* **10 (3)**, 1369 (2012).

[Delgado2020] R. Delgado-Buscalioni, A. Vázquez-Quesada, M. Meléndez: *In preparation* (2020).

[Johannsmann2008] D. Johannsmann: *Viscoelastic, mechanical and dielectric measurements on complex samples with the quartz crystal microbalance*, *Physical Chemistry Chemical Physics* **10**, 4516–4534 (2008).

[Johannsmann2009] D. Johannsmann, I. Reviakine, R. P. Richter: Dissipation in Films of Adsorbed Nanospheres Studied by Quartz Crystal Microbalance (QCM), Analytical Chemistry **81**, 8167–8176 (2009).

[Johannsmann2015] D. JOHANNSMANN: *The Quartz Crystal Microbalance in Soft Matter Research: Fundamentals and Modeling*. Springer (2015).

[Melendez2020] M. MELÉNDEZ, A. VÁZQUEZ-QUESADA, R. DELGADO-BUSCALIONI: *Load impedance of an immersed extended layer on the quartz crystal microbalance*, *in preparation* (2020).

[Nomura1982] T. NOMURA, and M. OKUHARA: *Frequency shifts of piezoelectric quartz crystals immersed in organic liquids*, *Analytica Chimica Acta* **142**, 281–284 (1982).

[Kanazawa1985] K. KANAZAWA, and J. GORDON II: *The oscillation frequency of a quartz resonator in contact with a liquid*, *Analytica Chimica Acta* **175**, 99–105 (1985).

[Peskin2002] CH. PESKIN: *The immersed boundary method*, *Acta Numerica*, pp. 479–517 (2002).

[Sauerbrey1959] G. SAUERBREY: *Verwendung von Schwingquarzen zur Wägung dünner Schichten und zur Mikrowägung*, *Zeitschrift für Physik* **55**, 206–222 (1959).