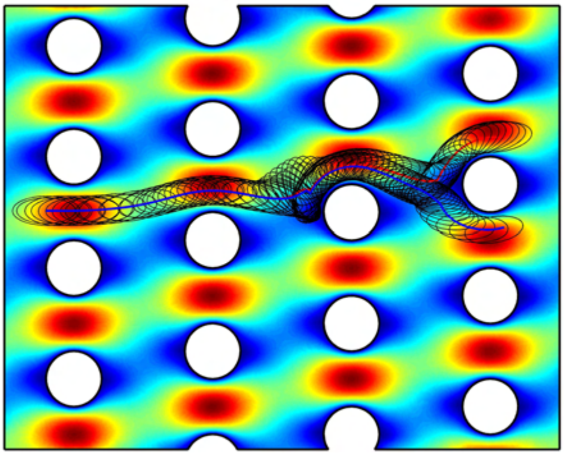
Master Thesis, s001396

Dynamics of finite-sized particles in microfluidic systems

Martin Heller



Supervisor: Henrik Bruus

MIC – Department of Micro and Nanotechnology
Technical University of Denmark

July 31, 2005

Abstract

Due to the advances in microfluidics and its applications in lab-on-a-chip systems the demand for numerical predictions of the behavior of flows on the microscale is growing. The aim of many existing microfluidic systems is handling and analyzing of various types of particles. The study of the dynamics of particles in microsystems is challenging because particles in microsystems often are of a size comparable to the dimensions of the containing microchannels.

Several methods for studying the behavior of particles in fluidic systems numerically exist. However, most of these methods rely on the particles being small so the can be regarded as point particles. Some models take the finite size of the particles into consideration by adding corrections to the flow in a region near the point particle. Some models exist that models particles of finite size, but these models rely on the particles being spherical [19].

In this thesis we develop a numerical code for simulation of the dynamics of finite-sized rigid particles of arbitrary shape. The model takes interactions between the particles and solid walls into account.

We test the implementation by simulating the convection of particles of different sizes and shapes by a viscous fluid through a microfluidic system consisting of an array of obstacles.

Accurate computation of the forces affecting the particles and precise resolution of the flow near the particle is crucial for the precise determination of the path of micrometer sized particles in microchannels. The simulation results presented in this thesis compare favorably with the observed behavior of particles in physical bumper arrays.

Resumé

Grundet udviklingen inden for mikrofluidik og dets anvendelse i “lab-on-a-chip”-systemer er der et stigende behov for numeriske forudsigelser af strømninger i mikroskalastørrelse. Formålet med mange eksisterende mikrofluide systemer er håndtering og analyse af forskellige partikler. Studiet af partiklers dynamik i mikrosystemer er vanskelig, fordi partikler ofte har en størrelse, der er sammenlignelig med dimensionerne af de mikrokanaler, hvori partiklerne transporteres.

Der findes flere metoder til numerisk undersøgelse af partiklers opførsel i væskesystemer. De fleste af disse metoder bygger på en antagelse om, at partiklerne kan betragtes som punktpartikler. Enkelte modeller tager partikler af en endelig størrelse i betragtning ved at tilføje korrektioner til væskestrømningen i et område i nærheden af partiklerne. Der findes modeller, som simulaere partikler af finitte dimensioner, men disse er baseret på, at partiklerne er kugleformede [19].

I denne afhandlingsopgave udvikler vi en computerkode til at simulere dynamikken for hårde partikler af vilkårlig facon. Modellen tager interaktioner mellem partiklerne og kanalernes massive vægge med i betragtning.

Implementationen testes ved at simulere transporten af partikler med forskellig størrelse og form. Partiklerne transporteres af en viskøs væske gennem et mikrosystem med en serie af forhindringer.

Præcis beregning af de kræfter, som påvirker partiklerne, samt nøjagtig gengivelse af strømningen i nærheden af partiklerne er helt afgørende for den præcise bestemmelse af banerne for mikrometer store partikler i mikrokanaler. Resultaterne fra simuleringerne i denne afhandling er sammenlignelige med den observerede opførsel af partikler i fysiske “bumper arrays”.

Preface

The present master thesis is submitted in candidacy to the cand.polyt. title at DTU – Technical University of Denmark. The project is part of the requirements for obtaining a MSc degree in technical physics with a specialization in “Biophysics and Complex Systems”.

The duration of the project has been from August 2004 to July 2005. The project has been carried out at the Bio/Chemical Microsystems (BCMS) section of MIC – Department of Micro and Nanotechnology in the Microfluidics Theory and Simulation Group (MIFTS). Some of the results have been presented at the 6$^{\text{th}}$ Liquid Matter Conference 2005 and in a paper submitted to Physical Review.

The work has been supervised by Prof. Henrik Bruus to whom I would like to express my gratitude for his inspiration, help and guidance provided during the compilation of the thesis. I should also like to thank MSc Jason Beech and Dr. Jonas Tegenfeld from Lund University for inspiring discussions and exchanged experimental data.

Finally I would like to acknowledge the members of the MIFTS group for their support. Especially the PhD students Mads Jakob Jensen and Laurits Højgaard Olesen for their unlimited patience and competent introduction to Matlab and Femlab during the first part of my studies.

Martin Heller
MIC – Department of Micro and Nanotechnology
Technical University of Denmark
July 31, 2005

Contents

List of Tables

List of Figures

List of Symbols

Physical quantities

Symbol	Description	Unit
$\boldsymbol{n}$	Unit normal vector	1
L_{diff}	Diffusion length	m
D	Diffusion constant	$\text{m}^2 \cdot \text{s}^{-1}$
t	Time	s
λ	Center-to-center distance between bumpers	m
N	Number of rows in a period of a bumper array	
d	Width of the gap between two posts	m
Re	Reynolds number	
Ca	Capillary number	
$\boldsymbol{r}$	Position vector	m
x, y, z	Cartesian coordinates	m
$\boldsymbol{u}$	Velocity field of a fluid	$\text{m} \cdot \text{s}^{-1}$
ρ	Density	$\text{kg} \cdot \text{m}^{-3}$
η	Viscosity	Pa·s
p	Hydrostatic pressure	Pa
$\boldsymbol{\sigma}$	Stress tensor	$\text{N} \cdot \text{m}^{-2}$
M	Molar mass	$\text{kg} \cdot \text{mol}^{-1}$
N_{A}	Avogadro's number	mol^{-1}
V	Volume	m^{D}, $\text{D} = 2, 3$
$\boldsymbol{a}$	Area vector	$\text{m}^{\text{D}-1}$, $\text{D} = 2, 3$
$\boldsymbol{f}$	Volume force	$\text{N} \cdot \text{m}^{-\text{D}}$, $\text{D} = 2, 3$
$\boldsymbol{F}$	Force	N
$\boldsymbol{\Omega}$	Angular velocity	$\text{rad} \cdot \text{s}^{-1}$
$\boldsymbol{\omega}$	Angular velocity	$\text{rad} \cdot \text{s}^{-1}$
$\boldsymbol{\theta}$	Rotation angle	rad
$\boldsymbol{\tau}$	Torque	N·m
Γ	Particle-liquid interface in level-set formulation	
P	Domain inside particles	
∂P	Particle surface	
$\boldsymbol{v}$	Translation velocity of a particle	$\text{m} \cdot \text{s}^{-1}$
m	Mass	kg
r_c	Critical radius	m
I	Moment of inertia for 2D particle	$\text{kg} \cdot \text{m}^2$
$\boldsymbol{R}$	Position vector from center of mass of a particle	m
R	Radius of a particle	m
Q	Flow rate	$\text{m}^{\text{D}} \cdot \text{s}^{-1}$, $\text{D} = 2, 3$
λ_1 and λ_2	Surface force on solid particles (2D)	$\text{N} \cdot \text{m}^{-1}$
ψ, ϕ	Level set functions	

Mathematical symbols

Symbol	Description	Unit
∂	Partial differential operator	
d	Ordinary differential operator	
D	Substantial derivative	
$\boldsymbol{\nabla}$	The del operator	m^{-1}
$\cdot$	Scalar product operator	
$\times$	vector product operator	
Ω	Computational domain	
$\partial\Omega$	Boundary of computational domain	
ν_i	Basis function on node i	
μ_i	Lagrange multiplier	
G	Generalized Neumann boundary condition in Femlab	
R_i	Dirichlet boundary condition in Femlab	
$\boldsymbol{U}$	Femlab solution vector	
$\boldsymbol{\Gamma}$	Femlab flux vector	
d_a	Femlab coefficient	
$\tilde{}$	Denotes dimensionless quantities	
$\hat{}$	Denotes the hat vector[1]	
H	Heaviside function	
δ	Kronicker's delta function	m^{-1}

List of Femlab code

[1] Danish: tværvektor

Chapter 1

Introduction

1.1 Microtechnology

“There’s Plenty of Room at the Bottom”. These were the words of Richard P. Feynman in his famous speech from 1959 [11]. This speech inspired scientists to investigate the physics and chemistry of systems with characteristic length scales ranging from a few nanometers to 100 µm. The first branch of technology that benefitted from the miniaturization of the physical systems was the electronics industry.

Using the many powerful fabrication techniques originally developed for the use in microelectronics new microsystems are being engineered. The diversity is large ranging from small mechanical systems, optical devices and magnetic elements to systems for handling of fluids. In recent years the fabrication techniques have been improved to a stage where fabrication of complex microsystems with several integrated components of various types has become possible [21]. Among this type of systems are systems known as lab-on-a-chip systems or micro total analysis systems (µTAS).

The medical and biological industries show growing interest for microsystems for point of care investigation of biological samples. This has given rise to an increasing need for the investigation of the physics of systems capable of handling this type of tasks.

Many biological applications involve analysis of aqueous solutions. Many of the fluids that are interesting in a biological context have very complex behavior. Blood for example is a non-Newtonian fluid changing its viscosity as a function of applied shear. Other fluids in the biological field contain salts or particles (cells) that cause nontrivial phenomena to occur.

In this thesis we will look closer into the behavior of fluidic microsystems for handling of fluids containing particles of finite size. Here finite size means that the typical length scale of a particle is comparable with the size of the microchannels containing it.

1.2 Microfluidics

When miniaturizing a fluidic system the relative strength (importance) of the involved forces change. In macroscopic systems the inertial forces dominate over viscous forces, and volume effects dominate over surface effects. In microsystems it is the other way round. Here viscosity is the dominating force, and hence most flows are laminar. This allows transport of different fluids side by side in the same channels without mixing. The dominance of surface forces over volume forces makes capillary effects important and the small size of the systems makes diffusion important.

Microfluidics is about taking advantage of these characteristic dominating forces and effects. One area where microfluidic systems have proven to be especially useful is the area of particulate flows. In this field handling and investigation of cells and other biomolecules have been given special interest. Microfluidic systems have been developed for many different tasks, but in this thesis we will focus on devices for sorting micrometer sized particles in microsystems.

1.3 Sorting devices

Several methods for sorting of particles are possible. Of course, the method of choice for a given application depends on the specific nature of the problem. In the following sections we will consider methods for sorting of particles based on their size. We will briefly mention some of the conventional sorting methods and introduce the concept of bumper arrays, which is a new class of sorting devices for fast deterministic separation of particles in microfluidic systems. We will only consider passive devices, *i.e.*, devices with no moving parts and no sensing equipment.

Diffusion based separation

Diffusion-based separation techniques rely on the statistics of diffusion distances. The average length L_{diff} covered by a molecule during the time t is given by

$$L_{\mathrm{diff}} = \sqrt{Dt}, \tag{1.1}$$

where D is the diffusion constant. The diffusion constant is related to the size of the particle, and smaller particles normally have larger diffusion constants. Hence, during a certain time interval small particles will on average travel a longer distance than large particles. This is the basic principle of diffusion based separation [25].

In figure 1.1 we sketch a simple device for separating small particles from a sample containing particles of various sizes. Real devices can of course be more sophisticated, but they all rely on the laminar flows in the micro regime and on the size dependence of diffusion.

The improvement of diffusion based sorting devices often involves some kind of symmetry breaking. The source of the symmetry breaking can be manyfold, *e.g.* electrical fields or perturbations of the channel geometry. A famous example is the rachet type of diffusion based separation devices [8, 9, 13, 15]. Here the diffusion is made anisotropic by introducing obstacles in the geometry. The obstacles affect small and larger particles differently thus improving separation.

Deterministic separation

One of the major drawbacks of diffusion based separation is that diffusion is a process stemming from the random thermodynamic movement of free particles (Brownian motion). This means that the separation is not exact, but rather one obtains a distribution of particle

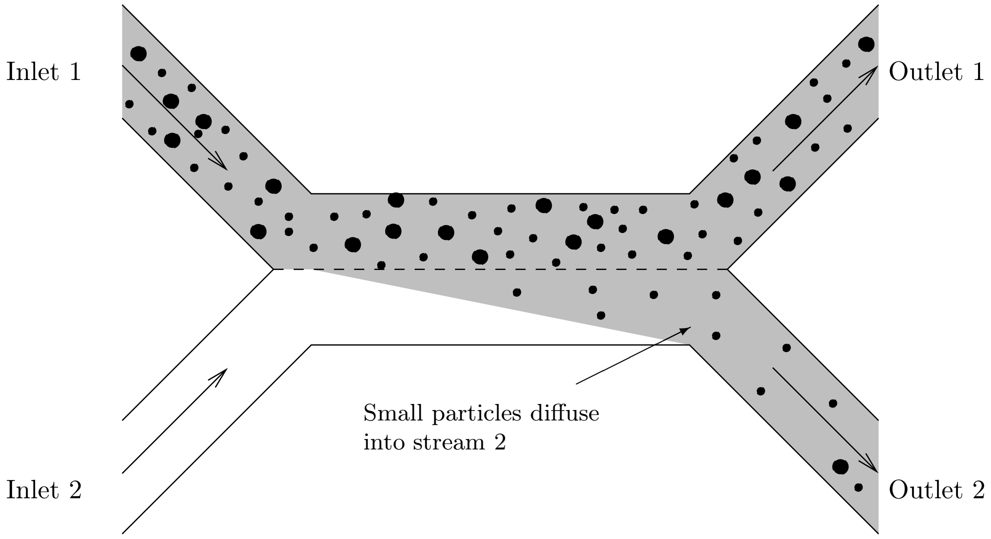


Figure 1.1: The basic principle of diffusion-based separation devices rely on the laminar flows of fluids in microsystems and on the size dependence of the diffusion constant. This simple device has two inlets. In Inlet 1 a fluid containing particles of different sizes enters and meets the fluid coming from Inlet 2 in a laminar nonmixing stream indicated by the dotted line. The smaller particles from stream 1 can diffuse across this separation line as indicated by the shaded area.

sizes corresponding to the probability of a particle of a given size to diffuse into the collecting stream. A more attractive separation scheme would be a deterministic process that was not limited by diffusion.

Such a device was suggested by Huang *et al.* [14]. They presented a microfluidic device, the bumper array, that made fast deterministic separation of particles possible at the microscale. A bumper array consists of a periodic array of posts situated a distance λ apart. Each row of posts is displaced a small amount $\Delta\lambda$ with respect to the previous row, see figure 1.2(a).

Because of the laminar flow in the microregime, the flow will be divided into a number of lanes corresponding to the periodicity of the array, see figure 1.2(b). If the particles are sufficiently small they will simply follow the flow in a zig-zag path through the bumper device. If, however, the particles are larger than a certain critical size they will be displaced from one lane into the neighboring one because they “bump” into the obstacles. This is called the displacement path and is sketched in figure 1.2(c). The critical size of the particles is determined by the geometry of the array as we will return to in chapter 4.

Bumper arrays do not to the same extend as conventional separation devices suffer from broadening due to diffusion. This is due to the fact that particles in a bumper array are only allowed to travel a short distance before their path is ‘reset’ due to contact with an obstacle. Small particles need to diffuse more than the width of one of the flow lanes in order to change migration mode. Thus in a bumper array the quality of the separation improves with the flow rate allowing fast separation.

Because of these desirable properties of the bumper arrays, it is worthwhile to examine what determines the critical size of the particles. This could lead to improved devices or easier integration into existing lab-on-a-chip systems. Another interesting motivation for the study of bumper arrays is the desire to separate particles of approximately equal hydrodynamic size deterministically based on other properties than their size. One possible separation parameter could be the morphology of the particles, and in chapter 6 we present what to our knowledge is the first analysis of this separation method. Morphological separation devices could have important medical applications. Many biomolecules, *e.g.* vira, are of similar size which makes it difficult to separate them in existing microsystems. However, the shapes of biomolecules are often quite well defined and distinct which makes morphological separation valuable.

In this thesis we will examine the dynamics of finite-sized particles in microchannels and develop a numerical code for future studies of the dynamics of arbitrarily shaped particles in microfluidic systems.

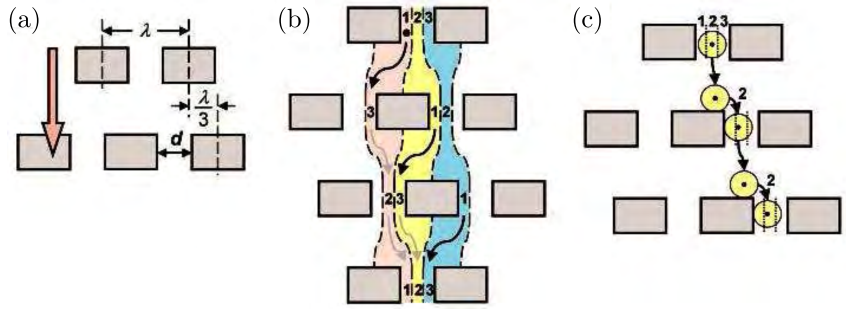


Figure 1.2: (a) A bumper array consists of a periodic array of obstacles. Each row is displaced a small amount $\Delta\lambda$ with respect to the previous row. (b) Because of the laminar flow in the microfluidic regime and the periodicity of the array, the flow in a bumper array consists of N lanes each carrying $1/N$ of the flow. (c) Small particles will just be passively convected by the flow while particles above a certain critical size will be displaced due to collisions with the obstacles. This causes particles of different size to migrate along different paths through the bumper array. The figure is reproduced from Ref. [14].

1.4 The structure of the project

The work carried out during this master project has fallen within several phases. The first phase of the project was devoted to learning the tools and methods necessary for working with computational fluidics. This involved in particular learning to use MATLAB, which the author had no priori experiences with, as well as getting acquainted with the scripting interface of FEMLAB. This involved reproducing a lot of known solutions to fluidic problems as well as a lot of experimentation with different equation solvers and setups.

When some familiarity with the use of MATLAB and FEMLAB was gained we started developing code for simulation of systems containing finite-sized particles. However, the computation times for finding the path of particles in even simple channel systems were huge.

The next part of the project was therefore devoted to finding alternative methods for simulating the systems of interest. We experimented with several methods, especially the Arbitrary Lagrange-Euler (ALE) method and the level set method. The level set method became the method of choice because it allowed us to simulate particle transport without the need to remesh during the computations. Furthermore, the level set method takes the interaction between the channel walls and the particles into account implicitly. We spent some time studying the method and finding an implementation in FEMLAB (see chapter 7). This part of the project was concluded with the submission of a paper to Physical Review E (see Appendix A) and the work was presented at the 6th Liquid Matter Conference (see Appendix B).

Due to limitations of the level set method the final part of the project was spent reworking some of the initial attempts of simulating particle transport. The experience gained from the work on the level set implementation allowed us to make a much more efficient implementation that enabled us to simulate particle transport in more complex systems with interactions between particles and channel walls.

This thesis documents the work carried out on simulating flows in two and three dimensions using FEMLAB and MATLAB as well as results obtained using both the level set method and direct simulation of the particle movement using new meshes each time step.

1.5 Notation and conventions

In this thesis we mainly use notation that follows ISO 31-11 [12]. That is, we use italic types for variables, such as the spacial dimensions x and y, for parameters, such as ρ or η, and for functions, such as $g(x)$. Vectors and tensors are denoted by bold face types, such as the the velocity vector $\boldsymbol{u}$ and the stress tensor $\boldsymbol{\sigma}$. Explicitly defined functions and operators are typeset in upright Roman type, *e.g.*, the trigonometric function cos or the differential operator d.

One important exception is the index notation which we occasionally use for brevity. In this notation, u_i is the i'th component of the vector $\boldsymbol{u}$. The index notation is also used for shorthand notation of partial derivatives where

$$\partial_t \phi(\boldsymbol{r}, t) \equiv \frac{\partial \phi(\boldsymbol{r}, t)}{\partial t} \tag{1.2}$$

is the partial derivative of the function $\phi(\boldsymbol{r}, t)$ with respect to t. We also use the Einstein convention for summation over indexes. Following this convention the scalar product between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ can be written as $u_i v_i$ where the repeated index means summation over that index.

Physical quantities consist of a numeral times a unit. These are typeset following the recommendations of ISO 1000: The units (and their prefixes) as well as the numbers are printed in upright Roman type. We use the SI base units as well as the SI derived units. Multiplication between numbers in the numerical value of a physical quantity is indicated by a '$\times$' while multiplication between units is indicated by a half-hight dot '$\cdot$'. The numerical value and the unit of a physical quantity are separated by a thin space, *e.g.*, a typical speed u of a flow in a microsystem is $u = 100 \times 10^{-6}\,\mathrm{m}{\cdot}\mathrm{s}^{-1} = 100\,\mu\mathrm{m}{\cdot}\mathrm{s}^{-1}$.

Chapter 2

Basic fluidics

It is not within the scope of this thesis to go through all aspects relevant for fluidics in microsystems in detail. In this chapter we will simply introduce basic concepts relevant for the problems under study in this thesis and state important equations governing the physics in fluidic systems. We will not cover every detail of the derivations but concentrate on the parts relevant for the implementation of the equations in numerical solvers. For a more thorough introduction to fluidics the reader is referred to [4, 5, 16–18].

2.1 Properties of fluids

Matter is usually classified as solids and fluids. Solids have a definite shape, and the atoms of the solid have fixed positions relative to one another. A solid will deform if a small external force is applied to it, but it will return to its initial configuration if the force disappears. If the applied force is larger than a certain threshold, the solid will deform plastically and obtain a new fixed configuration.

Fluids have no preferred configuration. Each element of a fluid can be moved freely around without affecting the macroscopic properties of the fluid. Two main groups of fluids exist: liquids and gasses. In a gas the intermolecular (or interatomic) distance is large, and the only interaction between the gas molecules is random collisions. The large interatomic distance makes a gas compressible.

On the other hand liquids are – like solids – ordered on the molecular scale and have an average intermolecular distance of the same order of magnitude as solids. However the atoms are not fixed on a lattice but can move freely. Because of the small intermolecular distance liquids can in most circumstances be regarded as incompressible. This is especially true when we consider liquids in microsystems, because the typical velocity of the liquid is orders of magnitude smaller than the velocity of sound. In this thesis we will therefore always assume the liquid to be incompressible.

This work will not be concerned with the properties of fluids on the atomic scale. Rather we are interested in the behavior of fluidic microsystems, *i.e.*, systems with characteristic length scales ranging from 1 µm up to a few millimeters. On such length scales the fluid can be seen as a continuum instead of a system consisting of individual atoms. This is called the *continuum hypothesis*. In other words, if we look at a small volume of liquid the physical quantities such as mass and momentum can be described as being uniformly distributed in this volume and are not properties of discrete points in space.

The validity of the continuum hypothesis can be verified by a small comparison between the characteristic length scales for the atomic scales of a fluid and the typical scales of the systems under consideration. The average interatomic spacing λ in a substance with molar mass M and density ρ is

$$\lambda = \left(\frac{M}{\rho N}\right)^{\frac{1}{3}}, \tag{2.1}$$

where N is Avogadro's number. For water we find

$$\lambda_{\mathrm{H_2O}} = \frac{18 \times 10^{-3}\,\mathrm{kg{\cdot}mol^{-1}}}{1 \times 10^{3}\,\mathrm{kg{\cdot}m^{-3}} \times 6.022 \times 10^{23}\,\mathrm{mol^{-1}}} \approx 0.31\,\mathrm{nm}. \tag{2.2}$$

Since the characteristic length scale for the microsystems under study is $L_0 \geq 1\,\mu\mathrm{m}$, the assumption $L_0 \gg \lambda$ holds true.

Thus we can describe the state of a liquid via continuous fields. In a given point in space $\boldsymbol{r}$ at a given moment in time t we can describe the fluid by the velocity $\boldsymbol{u}(\boldsymbol{r},t)$, the pressure $p(\boldsymbol{r},t)$, the density $\rho(\boldsymbol{r},t)$ and the viscosity $\eta(\boldsymbol{r},t)$. In many cases the density and the viscosity are constant in both time and space.

2.2 Governing equations

In the following we will derive the governing equations for a system consisting of a fluid flowing in a system consisting of channels with solid walls. The flow of the fluid governs the dynamics of solid finite-sized particles suspended in the fluid. Such a system is sketched in figure 2.1.

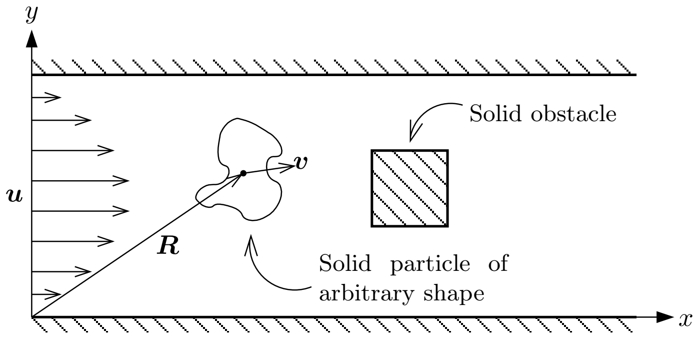


Figure 2.1: Sketch of a typical microfluidic system with finite-sized particles. The figure shows a microfluidic system for handling of finite-sized particles of arbitrary shape. The size of the particles is comparable to the size of the channels, and the particles can therefore not be described as point particles.

The fluid

If we consider a small volume of fluid $\mathrm{d}V$, the mass of the fluid contained in this volume is $\int \mathrm{d}V\,\rho$. The flux of mass through an area element $\mathrm{d}\boldsymbol{a}$ of the test volume is naturally

$\rho\boldsymbol{u}\cdot\mathrm{d}\boldsymbol{a}$. Therefore the total mass leaving the unit volume in unit time is the closed surface integral over the entire surface $\partial\Omega$ of the test volume Ω

$$\oint_{\partial\Omega}\rho\boldsymbol{u}\cdot\mathrm{d}\boldsymbol{a}. \tag{2.3}$$

The amount of mass leaving the volume must obviously equal the decrease in mass inside the volume per unit time

$$-\frac{\partial}{\partial t}\int_{\Omega}\mathrm{d}V\,\rho. \tag{2.4}$$

Thus in order to conserve the total mass of the system we must have the equality

$$-\frac{\partial}{\partial t}\int_{\Omega}\mathrm{d}V\,\rho=\oint_{\partial\Omega}\rho\boldsymbol{u}\cdot\mathrm{d}\boldsymbol{a}. \tag{2.5}$$

The right hand side of this equation can be rewritten using Gauss' theorem

$$-\frac{\partial}{\partial t}\int_{\Omega}\mathrm{d}V\,\rho=\int_{\Omega}\mathrm{d}V\,\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}). \tag{2.6}$$

Collecting all terms in a common integral yields

$$\int_{\Omega}\mathrm{d}V\left[\frac{\partial\rho}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u})\right]=0, \tag{2.7}$$

which must hold true for any volume in the fluid, and we can thus conclude that

$$\frac{\partial\rho}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u})=0. \tag{2.8}$$

If the fluid is incompressible, the density of the fluid is constant, and the above equation simplifies to

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0, \tag{2.9}$$

which is the *continuity equation* for incompressible fluids.

Now we need an equation that describes the evolution of the velocity field. We would like an equation that gives us the velocity of the fluid at a given point in space as a function of time, *i.e.*, in an Eulerian reference frame. Applying Newton's second law on an infinitesimal volume of the fluid gives

$$\rho\,\mathrm{D}_t\,\boldsymbol{u}=\sum_i\boldsymbol{f}_i, \tag{2.10}$$

where D_t is the substantial differential operator and $\boldsymbol{f}_i$ is the volume forces working on the unit volume of the fluid. Expanding the left hand side of the equation to ordinary differential operators we find

$$\rho\Big(\partial_t\boldsymbol{u}+(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\Big)=\sum_i\boldsymbol{f}_i. \tag{2.11}$$

The forces on the right hand side of the equation can be any external force working on the fluid, *e.g.* the gravitational force, electrical forces or magnetic forces. However, in this thesis we will only consider pressure forces and viscous forces.

The forces due to a pressure p working on a small volume of fluid $\mathrm{d}V$ are equal to the surface integral of the inward pointing normal vector $-\boldsymbol{n}$ times the pressure

$$\boldsymbol{F}_{\text{pressure}} = \int \mathrm{d}a\,(-\boldsymbol{n}p). \tag{2.12}$$

Using Gauss' theorem we can convert this surface integral to a volume integral

$$\boldsymbol{F}_{\text{pressure}} = -\int \mathrm{d}V\,(\boldsymbol{\nabla}p). \tag{2.13}$$

from which it is apparent that a pressure force

$$\boldsymbol{f}_{\text{pressure}} = -\boldsymbol{\nabla}p \tag{2.14}$$

works on any unit volume $\mathrm{d}V$ of the fluid.

The viscous forces in fluids are due to internal friction. It is therefore apparent that the viscous momentum transfer in the fluid can only occur if the different parts of the fluid move with relative velocities with respect to one another. Or in other words: the viscous stress must depend on the spacial derivatives of the velocity.

In microfluidic systems the flow is typically laminar, and the gradients in the velocities are small. We can therefore assume that the viscous stress is only depending on the first order derivatives $\partial u_i/\partial r_k$ of the velocity. In particular we know that no terms can be independent of $\partial u_i/\partial r_k$ since the viscous stress must vanish if the velocity of the fluid is constant.

Also, there can be no viscous stress in the fluid if it is in uniform rotation, *i.e.*, if the velocity equals

$$\boldsymbol{u} = \boldsymbol{\Omega} \times \boldsymbol{r}, \tag{2.15}$$

where $\boldsymbol{\Omega}$ is the angular velocity of the fluid. The above assumptions are satisfied by the tensor

$$\sigma'_{ik} = \eta\left(\frac{\partial u_i}{\partial r_k} + \frac{\partial u_k}{\partial r_i} - \frac{2}{3}\delta_{ik}\frac{\partial u_j}{\partial r_j}\right) + \zeta\delta_{ik}\frac{\partial u_j}{\partial r_j}, \tag{2.16}$$

where η and ζ are the first and second viscosity coefficients, respectively. Since we are only considering incompressible fluids in this thesis, we can use the incompressibility condition derived above to simplify the viscous stress tensor

$$\sigma'_{ik} = \eta\left(\frac{\partial u_i}{\partial r_k} + \frac{\partial u_k}{\partial r_i}\right). \tag{2.17}$$

The viscous stress tensor gives us the i'th component of the friction force per unit area acting on the surface element $\mathrm{d}a$ with surface normal parallel to the k'th unit vector. Thus we can write the viscous surface force on a unit volume of the fluid as

$$\mathrm{d}F_i = \sigma'_{ik} n_k\,\mathrm{d}a. \tag{2.18}$$

The total viscous force on a unit element of the fluid can therefore be obtained by integrating over the surface of the element

$$\big(\boldsymbol{F}_{\text{viscous}}\big)_i = \int_{\partial\Omega} \mathrm{d}a\,\sigma'_{ik} n_k = \int_{\Omega} \mathrm{d}\boldsymbol{r}\,\partial_k\sigma'_{ik}, \tag{2.19}$$

where the last equality is obtained using Gauss' theorem. From the above equation we realize that the viscous force per unit volume is simply the gradient of the viscous stress tensor. As was shown in Eq. (2.14), the pressure force is also a gradient. It is therefore natural to combine the pressure force and the viscous force into a single tensor, the stress tensor, defined by

$$\sigma_{ik} = -p\delta_{ik} + \sigma'_{ik}. \tag{2.20}$$

Using this tensor the total force per unit volume due to pressure and viscosity can be written as

$$\Big(\boldsymbol{f}_{\text{pressure}} + \boldsymbol{f}_{\text{viscous}}\Big) = \partial_k \sigma_{ik}. \tag{2.21}$$

Since the pressure force and the viscous force are the only two forces acting on the fluidic systems under considerarion in this thesis, the sum of forces in Eq. (2.11) consists only of this term. Substituting the force term into Eq. (2.11) yields

$$\rho\Big(\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\Big) = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}, \tag{2.22}$$

or if we carry out the multiplication on the right hand side

$$\rho\Big(\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\Big) = -\boldsymbol{\nabla}p + \eta\boldsymbol{\nabla}^2\boldsymbol{u}. \tag{2.23}$$

This equation is the Navier-Stokes equation which governs the motion of incompressible fluids in microfluidic systems. For time-independent systems, the time derivative of the velocity field is zero and the equation reduces to

$$\rho(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla}p + \eta\boldsymbol{\nabla}^2\boldsymbol{u}. \tag{2.24}$$

The systems under study in this thesis consist of closed channels with solid walls and solid obstacles. It is an experimentally established fact that the fluid velocity at a solid interface $\partial\Omega$ equals the velocity of the interface. This is known as the *no-slip boundary condition.* In most cases we will use the containing channels as the reference frame. In that case the no-slip condition reads

$$\boldsymbol{u} = \boldsymbol{0}, \quad \boldsymbol{r} \in \partial\Omega. \tag{2.25}$$

In general it is not possible to solve the derived equations analytically, and one has to rely on numerical methods. However, it is useful to test the validity of the numerical results by comparing numerical results with the analytical solution to some of the few analytically solvable configurations.

Suspended particles

Particles in a fluid move according to Newton's laws. In general, if we consider a rigid particle P, it translates with a velocity $\boldsymbol{v}$ given by

$$\boldsymbol{v} = \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} \tag{2.26}$$

$$m_{\text{part}}\frac{\mathrm{d}^2\boldsymbol{r}_{\text{part}}}{\mathrm{d}t^2} = \sum_i \boldsymbol{F}_i, \tag{2.27}$$

where $\boldsymbol{r}_{\text{part}}$ is the position of the center of mass for the particle and ρ_{part} is the density of the particle. The body forces $\boldsymbol{F}_i$ working on the particles can of course originate from a variety of sources depending on the particle properties. Many microfluidic systems for particle handling utilize the possibility of manipulating particles with external forces such as magnetic forces, (di)electrophoretic forces etc. In the systems under investigation in this thesis, however, two force terms are of interest: The hydrodynamic forces $\boldsymbol{F}_{\text{fluid}}$ exerted by the fluid and contact forces $\boldsymbol{F}_{\text{contact}}$ between solid obstacles and the particles.

The hydrodynamic forces can be calculated explicitly by integrating the viscous stress tensor $\boldsymbol{\sigma}$ over the surface ∂P of the particle

$$\boldsymbol{F}_{\text{fluid}} = \int_{\partial P} \mathrm{d}a\, \boldsymbol{\sigma} \cdot \boldsymbol{n}, \tag{2.28}$$

where $\boldsymbol{n}$ is a unit normal vector to the surface of the particle and $\mathrm{d}a$ is a unit area element.

The contact forces on the contrary can not be calculated explicitly. They are discrete forces working only when collisions occur. The physics of the collision forces depends on the properties of the colliding surfaces. Many different types of contact are observed: particles sticking to surfaces, particles sliding along the walls of the channels, repulsion or the particles could just be subjected to a normal force ensuring that the particles do not move into the walls.

In the models in this thesis we will model the contact force as a repulsive force working on the surface of the particles. The repulsion is always orthogonal to the walls and is smeared out over a short distance away from the wall. We will return to the model of the contact force when we describe the implementation of the equations in chapter 6.

If the suspending flow is not completely uniform the particle will rotate about an axis through its center of mass with the angular velocity $\boldsymbol{\omega}$ because of the torque $\boldsymbol{\tau}$ given by

$$\boldsymbol{\tau} = -\int_{\partial P} \mathrm{d}a\, \boldsymbol{R} \times (\boldsymbol{\sigma} \cdot \boldsymbol{n}). \tag{2.29}$$

Above $\boldsymbol{R}$ is a vector pointing from the center of mass of the particle to the points on the particle perimeter. The angular velocity is found from the equation

$$I \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = \boldsymbol{\tau} + \boldsymbol{R} \times \boldsymbol{F}_{\text{contact}}, \tag{2.30}$$

where I is the moment of inertia for the particle. In the general three dimensional case the moment of inertia is a tensor. However, we consider only two dimensional particle transport in which case, the moment of inertia reduces to $I = \int_P \mathrm{d}V\, \rho_{\text{part}} |\boldsymbol{R}|^2$, where $\boldsymbol{R}$ is a vector in the plane of the motion and the axis of rotation is perpendicular to this plane.

The motion of the particle is coupled to the fluid flow by a no-slip condition on the surface of the particle

$$\boldsymbol{u}(\boldsymbol{r}) = \boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{R} \quad , \boldsymbol{r} \in \partial P. \tag{2.31}$$

2.3 Forces on particles

In the following we will consider some results relating to particles/obstacles in a flow. Later we will use these results to validate the numerical code. Furthermore, some analytical results for particles in a flow will be derived in order to get an understanding of the general principles underlying particle motion in a flow.

As mentioned, it is not possible to solve Navier-Stokes equation and the continuity equation in general, but analytical results can be obtained in simple geometries or in regimes where some terms in the governing equations can be ignored. A famous result regarding particles in a fluid was obtained by Stokes. He derived an expression for the drag force F_{drag} working on a spherical particle submersed in an infinite fluid. The result can be found in practically any text book on fluidics and reads

$$F_{\text{drag}} = 6\pi\eta a U_0, \tag{2.32}$$

where η is the viscosity of the fluid, a is the radius of the sphere and U_0 is the velocity of the fluid far away from the sphere. The result is only valid in the Stokes regime (low Reynolds number) and if the flow field is homogeneous far away from the spherical particle.

This result has often been used to model the dynamics of particles in a flow. Either one can solve the flow problem first and then use Eq. (2.32) to find the path of the particle afterwards. This is known as a one-way coupling because the motion of the particle is not affecting the flow of the fluid. Another possibility is a two-way coupling where Eq. (2.32) is added as a point source in Navier-Stokes equation, and the motion of the particle and the flow field of the fluid are solved for simultaneously.

Neither of these two methods are applicable for the type of problems sketched in figure 2.1 because they neglect to take the finite size of the particles into account and the Stokes drag formula is only valid in uniform flows which is not the case when the particles are of finite size. Furthermore the orientation of the particles are not accounted for. To account for these effects, as well as the nonlinear effects arising from the discrete collision forces, the finite size of the particles must contribute to the solution of the flow problem. In the following chapters we will look closer into a numerical solution to this problem. First, however, we will consider some qualitative results relating to finite-sized particles in shear flows and in the vicinity of solid walls in order to know what is to be expected from the simulations.

Particle in a shear flow

If a finite-sized particle is convected by a nonhomogene flow, there will be a relative difference between the flow velocity $\boldsymbol{u}$ and the translation velocity $\boldsymbol{v}$ of the particle because of the finite size of the particle (see figure 2.2). This shear induced relative difference in velocity will cause the particle to rotate. Furthermore, if the relative difference in velocities is larger on one side of the particle than on the other, this will induce an uneven pressure distribution leading to a velocity component perpendicular to the translation direction. This is known as the Saffman effect.

The details of the shear induced rotation and translation are naturally depending on the shape of the particle. Because of the working principle of the bumper arrays, our model needs to take these effects into account, especially if we want to consider the paths of particles of arbitrary shape. Therefore we can not use the simple approach outlined above but we need to explicitly determine the forces on the particles to take the finite size of the containing geometries into account. In chapter 3 we will describe how we implement a direct and accurate calculation of the forces exerted on the particles.

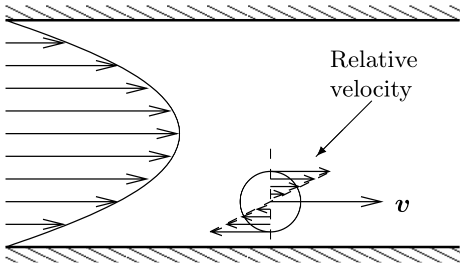


Figure 2.2: A particle of finite size convected with the translation velocity $\boldsymbol{v}$ by a flow with a nonuniform velocity profile $\boldsymbol{u}$ will experience a flow relative to the particle. This relative flow can induce rotation and a transverse velocity component due to an uneven pressure distribution on the surface of the particle.

Particle near a solid wall

In the vicinity of solid walls the flow is not uniform. This causes perturbations in the force experienced by the particles. In microsystems it is often the case that the size of the particles is comparable to the size of the channels in which the particles are contained. Therefore particles in microsystems seldom experience uniform flow fields.

We will consider the flow between a solid wall and a particle translating with velocity $-U$ relative to the wall in close vicinity of this wall (see figure 2.3).

If $\alpha\frac{\rho dU}{\eta} \ll 1$, where ρ is the density of the fluid, d is the gap height and η is the viscosity of the fluid, then we are in the lubrication limit. Here the velocity profile in the gap between the particle and the solid wall can be described as the superposition of a Poiseuille flow and a Couette flow [4]

$$u = -\frac{1}{2\eta}\frac{\mathrm{d}p}{\mathrm{d}x}y(d-y) + U\left(\frac{d-y}{d}\right). \tag{2.33}$$

Thus the flow rate Q per unit width of the fluid layer can be found by integration of u from

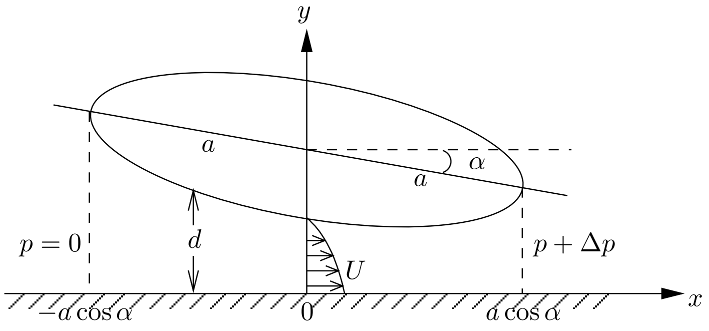


Figure 2.3: Sketch of an elliptical particle moving with a velocity $-U$ near a solid wall. The coordinate system is fixed with respect to the particle, giving rise to the slip velocity condition at the wall.

0 to $d(x)$

$$Q = \int_0^{d(x)} \mathrm{d}y\, u(x,y) = -\frac{1}{12\eta}\frac{\mathrm{d}p}{\mathrm{d}x}d^3 + \frac{1}{2}Ud \tag{2.34}$$

Isolating the pressure gradient from this expression yields

$$\frac{\mathrm{d}p}{\mathrm{d}x} = 6\eta\left(\frac{U}{d^2} - \frac{2Q}{d^3}\right). \tag{2.35}$$

If we choose the pressure to be zero at $x = -a\cos\alpha$ we can solve for the pressure $p(x)$

$$p(x) = 6\eta U \int_{-a\cos\alpha}^{x} \mathrm{d}x\, \frac{1}{d^2(x)} - 12Q\eta \int_{-a\cos\alpha}^{x} \mathrm{d}x\, \frac{1}{d^3(x)} \tag{2.36}$$

$$= 6\eta U g_1(x) - 12Q\eta g_2(x), \tag{2.37}$$

where

$$g_1(x) = \int_{-a\cos\alpha}^{x} \mathrm{d}x\, \frac{1}{d^2(x)} \quad \text{and} \quad g_2(x) = \int_{-a\cos\alpha}^{x} \mathrm{d}x\, \frac{1}{d^3(x)} \tag{2.38}$$

are only depending on the geometry of the particle. Denoting the pressure difference from $x = -a\cos\alpha$ to $x = a\cos\alpha$ by Δp we find

$$\Delta p = p(a\cos\alpha) - p(-a\cos\alpha) = 6\eta U\Delta g_1 - 12Q\eta\Delta g_2, \tag{2.39}$$

where $\Delta p_i = p_i(-a\cos\alpha) - p_i(-a\cos\alpha)$, for $i = 1, 2$. From this equation we can determine the flow rate in the gap between the particle and the solid wall

$$Q = \frac{U}{2}\frac{\Delta g_1}{\Delta g_2} - \frac{1}{12\eta}\frac{\Delta p}{\Delta g_2}. \tag{2.40}$$

Inserting this expression back into the pressure function from Eq. (2.37) yields

$$p(x) = 6\eta U\left(g_1(x) - g_2(x)\frac{\Delta g_1}{\Delta g_2}\right) + \frac{\Delta p}{\Delta g_2}g_2(x). \tag{2.41}$$

From this expression we find that the pressure distribution on the particle near a solid wall originates from two sources: one local term proportional to the translation velocity of the particle and one term due to the global flow around the particle. We will use the result for the pressure distribution on particles near a solid wall to verify that the numerical code developed in chapter 3 gives the expected results.

Chapter 3

Finite element method

The finite element method is a technique for solving differential equations numerically. The method was originally developed for problems in structural mechanics, but due to the generality of the method, it is applicable for many types of problems. In this thesis we use the finite element method to solve hydrodynamic problems. We will not go into details with every aspect of the theory of the finite element method, but just state the background necessary to understand the implementation of the problems solved in this work.

The idea behind the finite element method is that a solution to a partial differential equation (PDE) problem can be approximated by a discretized solution with only a finite number of unknown parameters. The discretization is done on a mesh. In this thesis we use triangular mesh elements for two dimensional problems and tetrahedric elements for three dimensional problems.

The solution u to a PDE problem is then approximated as a linear combination

$$u(\boldsymbol{r}) = \sum_i U_i \nu_i(\boldsymbol{r}), \tag{3.1}$$

where U_i is the value of u at the i'th node point, and $\nu_i(\boldsymbol{r})$ is the i'th basis function. The basis functions are localized functions that take the value 1 in node i and 0 in every other node. The basis functions used in this thesis are of a type known as Lagrange elements, *i.e.*, they are piecewise polynomials of degree k. We use only linear elements $k = 1$ and quadratic elements $k = 2$.

To exemplify how the finite element method is applied to hydrodynamic problems, we consider the timeindependent Navier-Stokes equation (2.24) derived in the previous chapter.

$$\rho(\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}. \tag{3.2}$$

This form of the equation is known as the strong form. It is only possible to satisfy this equation if we dispose over infinitely many basis functions. In practice this is not the case. Instead we multiply the equation with the basis function and integrate over the entire computational domain

$$\int_\Omega \mathrm{d}\boldsymbol{r}\, \nu_i \rho(\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = \int_\Omega \mathrm{d}\boldsymbol{r}\, \nu_i \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}. \tag{3.3}$$

This is known as the weak form of the equation because we solve the original problem only “on average”. If we perform partial integration on the right hand side of the above equation we obtain

$$\int_{\Omega} \mathrm{d}\boldsymbol{r}\, \nu_i \rho(\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = \int_{\Omega} \mathrm{d}\boldsymbol{r}\, \boldsymbol{\nabla}\nu_i \cdot \boldsymbol{\sigma} - \int_{\partial\Omega} \mathrm{d}a\, \nu_i \boldsymbol{n} \cdot \boldsymbol{\sigma}, \tag{3.4}$$

where we recognize the boundary integral as a Neumann type boundary condition. If the boundary condition is a Dirichlet constraint, this integral should just have the value required to fulfill this constraint. This can be achieved by introducing a generalized Neumann constraint defined as

$$-\, \boldsymbol{n} \cdot \boldsymbol{\sigma} = G + \frac{\partial R_i}{\partial u_i} \mu_i. \tag{3.5}$$

Substitution of the generalized Neumann condition into the weak equation yields

$$\int_{\Omega} \mathrm{d}\boldsymbol{r}\, \nu_i \rho(\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = \int_{\Omega} \mathrm{d}\boldsymbol{r}\, \boldsymbol{\nabla}\nu_i \cdot \boldsymbol{\sigma} + \int_{\partial\Omega} \mathrm{d}a\, \nu_i \left(G + \frac{\partial R_i}{\partial u_i} \mu_i \right), \tag{3.6}$$

where $0 = R_i$ is the Dirichlet condition and μ_i is a Lagrange multiplier. Exactly the same approach can be used to transform the continuity equation to the weak form.

3.1 FEMLAB

We have chosen to use the finite element solver FEMLAB for the numerical simulations done in this thesis [2]. The program has been chosen for several reasons. First of all it is the tool that is available at MIC. Secondly, FEMLAB is a general partial differential equation solver, and it is therefore possible to solve coupled problems involving many different physical fields. The latter will be important if we want to use our code to simulate the flow in real microfluidic devices. Such microchips may include electroosmotic pumps, magnetic elements, thermal components etc., all of which it is possible to implement in FEMLAB.

Another feature about FEMLAB is that it relies on the MATLAB scripting language. It is therefore possible to extend the built-in macros and functions from FEMLAB using MATLAB. Most of the scripts used in this work have been done in MATLAB.

3.2 Setting up the system in FEMLAB

FEMLAB *solves* all PDE problems as if they were weak problems, but we can give the equations in both strong and weak form depending on what suits our needs. It is beyond the scope of this thesis to describe in general how to set up a problem in FEMLAB. In the following section we will go through the steps we have taken to model a microfluidic system containing moving particles of arbitrary finite size and shape. The particle motion necessitates modeling of a system with a changing geometry and arcuate integration over particle boundaries in order to determine the forces correctly. Furthermore the geometry of the systems under study is periodic. This allowed us to reduce the size of the computational domain by implementing periodic boundary conditions instead of modeling the entire physical devices.

The changing geometry of the systems under study is in conflict with one of the limitations of FEMLAB: The computational domain in FEMLAB must be fixed. To cope with this problem, we implemented the timestepping via MATLAB using the following algorithm:

- First we define the geometry of the microsystem and solve the time-independent Navier-Stokes equation under the incompressibility condition for a given position and velocity of the particle.
- From this solution we calculate the forces and torque on the particle.
- Using the calculated forces and the torque we calculate the new position, velocity, orientation and rotation of the particle using Eqs. (2.26) and (2.30).
- We then update the position of the particle and construct a new mesh.
- Finally, we solve the equation system in the updated geometry using an interpolation of the old solution to the new mesh as the initial condition for the flow problem. The computed velocity and rotation of the particle is used as the boundary condition on the surface of the particle.

In the following sections we will go through the implementation of the fluidic problem in FEMLAB as well as the boundary conditions needed in our models. First we will describe how the governing equations are set up using the FEMLAB general form. Next we will go through the formulation of the boundary conditions. This involves defining weak expressions at the particle boundary.

Defining the equation system using the general form

In the general PDE form, FEMLAB accepts equations in strong divergence form

$$d_{a_i}\frac{\partial u_i}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{\Gamma}_i = F_i \qquad \text{in } \Omega, \tag{3.7a}$$

$$-\boldsymbol{n}\cdot\boldsymbol{\Gamma}_i = G_i + \frac{\partial R_j}{\partial u_i}\mu_j \qquad \text{on } \partial\Omega, \tag{3.7b}$$

$$0 = R_i \qquad \text{on } \partial\Omega, \tag{3.7c}$$

where the first equation is fulfilled in the computational domain Ω, and the latter two are the generalized Neumann and Dirichlet boundary conditions, respectively [3]. In the above equations $\boldsymbol{U} = u_i$ is the solution vector and t is the time. The coefficient d_{a_i} determines if the equations are time-dependent. FEMLAB satisfies the Dirichlet condition $R_i = 0$ in Eq. (3.7c) by choosing the necessary Lagrange multiplier μ_i. The coefficient G_i is the Neumann type boundary condition. All coefficients, including the flux vector $\boldsymbol{\Gamma}_i$ and the force term F_i, can be functions of the solution $\boldsymbol{U}$, its gradient $\boldsymbol{\nabla U}$ or space. If the problem is time-dependent, the coefficients can be functions of time or derivatives of the solution vector with respect to time as well. $\boldsymbol{n}$ is a normal vector.

For the flow problem we first need to put Navier-Stokes equation on a form suitable to enter into FEMLAB. As mentioned, we solve the problem as a stationary problem, *i.e.*, we can set $\partial_t\boldsymbol{u} = 0$ in Eq. (2.23). This reduces the equation to

$$\rho(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla}p + \eta\boldsymbol{\nabla}^2\boldsymbol{u}. \tag{3.8}$$

It turns out that it is a good idea to nondimensionalize the terms in the equation using characteristic scales for the system. Denoting the nondimensionalized variables with a tilde and the characteristic scales with the subscript 0 we have

$$u_0\tilde{\boldsymbol{u}} = \boldsymbol{u}, \quad L_0\tilde{\boldsymbol{r}} = \boldsymbol{r}, \quad \rho_0\tilde{\rho} = \rho, \quad \eta_0\tilde{\eta} = \eta, \tag{3.9}$$

where u_0, L_0, ρ_0 and η_0 are the characteristic scales for the velocity, length, density and viscosity, respectively. Using these characteristic scales, we can define a characteristic pressure scale as

$$p_0 = \frac{\eta_0 u_0}{L_0}. \tag{3.10}$$

If we substitute the physical parameters by their dimensionless counterparts multiplied by the characteristic scale, we obtain the Navier-Stokes equation in dimensionless form

$$Re\tilde{\rho}(\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\nabla}})\tilde{\boldsymbol{u}} = \Big(-\tilde{\boldsymbol{\nabla}}\tilde{p} + \tilde{\eta}\tilde{\boldsymbol{\nabla}}^2\tilde{\boldsymbol{u}}\Big). \tag{3.11}$$

Here Re is the Reynolds number defined as

$$Re = \frac{\rho_0 u_0 L_0}{\eta_0}. \tag{3.12}$$

To enter Eq. (3.11) into FEMLAB we have two opportunities. Either we put the pressure into $\boldsymbol{\Gamma}$, or we put the pressure gradient into $\boldsymbol{F}$. The first case is called the full stress tensor formulation while the latter is called the viscous stress tensor formulation. FEMLAB can solve the problem in either formulation, but the choice of formulation affects the boundary conditions as they depend on $\boldsymbol{\Gamma}$. We have chosen the full stress tensor formulation.

Rearranging the terms in the above equation and introducing the dimensionless stress tensor $\tilde{\boldsymbol{\sigma}} = -\tilde{p}\boldsymbol{I} + \tilde{\eta}\tilde{\boldsymbol{\nabla}}^2\tilde{\boldsymbol{u}}$, where $\boldsymbol{I}$ is a unit diagonal matrix, gives us the equation

$$\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{\sigma}} = Re\tilde{\rho}(\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\nabla}})\tilde{\boldsymbol{u}} \tag{3.13}$$

which is on the desired form with $\boldsymbol{\Gamma} = \tilde{\boldsymbol{\sigma}}$ and $\boldsymbol{F} = Re\tilde{\rho}(\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\nabla}})\tilde{\boldsymbol{u}}$. Because the problem is time-independent $\boldsymbol{d}_a$ is zero.

The dimensionless form of the continuity equation is obviously

$$0 = \tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{u}}. \tag{3.14}$$

Again we have two possibilities for putting it on a form suitable for FEMLAB. We choose to put the divergence of the velocity into the $\boldsymbol{F}$ term. In code example 1 is shown how the equations are implemented in a MATLAB script.

Boundary conditions

The boundary conditions are of course depending on the physics of the system we are modeling. In the following description we will formulate the boundary conditions needed for the model of the bumper arrays described in chapter 4. We need the following types

- No-slip velocity conditions at the channel walls and the obstacles.
- A pressure drop from the inlet to the outlet. Because of the periodicity of the bumper arrays, the velocity at the inlet must equal the velocity at the outlet.

```
% DIMENSIONS AND VARIABLES
fem.sdim        = {'x' 'y'};
fem.dim         = {'u' 'v' 'p' 'lm1' 'lm2'};
fem.shape       = {shlag(2,'u'),shlag(2,'v'),shlag(1,'p'),shlag(2,'lm1'),shlag(2,'lm2')};

% EQUATIONS
fem.equ.shape = {[1:3]};         % only use velocity and pressure in bulk
fem.form      = 'general';
fem.equ.f     = {{'Re*(u*ux+v*uy)';'Re*(u*vx+v*vy)';'ux+vy';'0';'0'}};
fem.equ.ga    = {{{'-p+2*eta*ux';'eta*(uy+vx)'};{'eta*(uy+vx)';'-p+2*eta*vy'}; ...
                  {'0';'0'};{'0';'0'};{'0';'0'}}};
```

Femlab code 1: Navier-Stokes equation and the continuity equation in Femlab notation in two spacial dimensions, `x` and `y`. `Re` is the Reynolds number, `eta` and `rho` are the viscosity and the density, respectively. `u` is the `x`-component of the velocity field, and `v` is the `y`-component of the velocity field. We use quadratic Lagrange elements for the velocity and linear elements for the pressure to avoid numerical instabilities in the pressure solution.

- At the surface of the particle we need to set the velocity of the fluid equal to the translation and rotation of the particle. At the same time we need to calculate the forces affecting the particle accurately.

In the following we will go through the implementation of each of these three boundary conditions. Although the derivation is for a two dimensional system, it is easily extendable to three dimensions. In the derivation we will use the velocity-pressure-force solution vector given by $\boldsymbol{U} = [u_1, u_2, p, \lambda_1, \lambda_2]$, where u_1 and u_2 are the x- and y-components of the velocity field, respectively. The pressure is given by p, and λ_1 and λ_2 are the x- and y-components of the forces on the particle surface. All variables are given in dimensionless coordinates.

With this notation the quantities $\boldsymbol{\Gamma}_i$ and F_i derived in the previous section become

$$\boldsymbol{\Gamma}_1 = \begin{bmatrix} \sigma_{11} \\ \sigma_{21} \end{bmatrix}, \quad \boldsymbol{\Gamma}_2 = \begin{bmatrix} \sigma_{12} \\ \sigma_{22} \end{bmatrix}, \quad \boldsymbol{\Gamma}_3 = \boldsymbol{\Gamma}_4 = \boldsymbol{\Gamma}_5 = \boldsymbol{0}, \tag{3.15a}$$

and

$$F_1 = Re\Big(\boldsymbol{u}\cdot\boldsymbol{\nabla}\Big)u_1, \quad F_2 = Re\Big(\boldsymbol{u}\cdot\boldsymbol{\nabla}\Big)u_2, \quad F_3 = \boldsymbol{\nabla}\cdot\boldsymbol{u}, \quad F_4 = F_5 = 0. \tag{3.15b}$$

The no-slip velocity condition

The no-slip velocity condition is a Dirichlet boundary condition, so we need to use Eq. (3.7c). The coordinate system is fixed with respect to the solid walls, so the velocity of the walls is zero. The boundary condition therefore reads

$$\boldsymbol{R} = \big[u_1, u_2, 0, 0, 0\big], \quad \text{on solid walls.} \tag{3.16}$$

If we choose $\boldsymbol{G} = \boldsymbol{0}$, we can insert the above choice of $\boldsymbol{R}$ into the generalized Neumann condition (3.7b) and obtain the equations

$$-\,\boldsymbol{n}\cdot\boldsymbol{\Gamma}_i = \mu_i, \quad \text{on solid walls.} \tag{3.17}$$

This does not impose any restrictions on the solution since Femlab is allowed to choose the Lagrange multipliers freely. Thus the velocity is constrained to zero at the solid walls as desired, and we have imposed no restriction on the pressure.

Inlet and outlet conditions

We want the velocity field to be periodic from the outlet to the inlet and a fixed pressure drop Δp between the two boundaries. The velocity is made periodic using the extrusion coupling variables and a linear mapping of the values of the velocity at the outlet boundary to the inlet boundary. To obtain this, we set no restrictions on the velocities at the outlet, *i.e.*, $\boldsymbol{R} = \boldsymbol{0}$. This gives us the Neumann condition

$$-\,\boldsymbol{n} \cdot \boldsymbol{\Gamma}_i = G_i. \tag{3.18}$$

Setting $G_3 = G_4 = G_5 = 0$ trivially fulfills this equation and imposes no constrictions on the pressure or the two force components. For the velocity components we have

$$-\,\boldsymbol{n} \cdot \begin{bmatrix} \sigma_{11} \\ \sigma_{21} \end{bmatrix} = G_1 \quad \text{and} \quad -\,\boldsymbol{n} \cdot \begin{bmatrix} \sigma_{12} \\ \sigma_{22} \end{bmatrix} = G_2. \tag{3.19}$$

Since we have chosen the full stress tensor formulation, the stress tensor components are functions of the pressure, and we have

$$-\,\boldsymbol{n} \cdot \begin{bmatrix} -p + 2\tilde{\eta}\,\partial_x u_1 \\ \tilde{\eta}(\partial_x u_2 + \partial_y u_1) \end{bmatrix} = G_1 \quad \text{and} \quad -\,\boldsymbol{n} \cdot \begin{bmatrix} \tilde{\eta}(\partial_x u_2 + \partial_y u_1) \\ -p + \tilde{\eta}\,\partial_y u_2 \end{bmatrix} = G_2. \tag{3.20}$$

If we set $G_i = \boldsymbol{n} \cdot p\boldsymbol{I}$, for $i = 1, 2$, this leaves us with a boundary condition that does not constrain the pressure and sets the viscous forces to zero at the inlet allowing for a periodic velocity gradient.

At the inlet we simply set the velocity equal to the velocity at the outlet by choosing the Dirichlet condition $\boldsymbol{R} = \left[u_1^{\text{out}} - u_1, u_2^{\text{out}} - u_2, (p^{\text{out}} + \Delta p) - p, 0, 0\right]$. The pressure drop is ensured by using the extrusion coupling variables denoted by the superscript "out" setting the pressure at the inlet equal to the pressure at the outlet plus the pressure difference. The Neumann coefficient is the same for the inlet as the one for the outlet because we do not want any viscous forces here either. This formulation ensures a pressure drop from the inlet to the outlet, but it does not fix the magnitude of the pressure. This is done by constraining the pressure to a fixed value (zero) at a point on the outlet boundary. If we want the pressure to be periodic as well, Δp is simply set to 0.

Boundary conditions at the surface of the particle

At the surface of the particle we must fulfill the no-slip condition (2.31). However, we need to determine the viscous forces at the particle boundary accurately. The viscous force is proportional to the gradient of the velocity field. Since we are using second order elements for the velocity, the gradients are only first order polynomials. Thus integrating the gradients at the particle boundary will not give us accurate results. We therefore use a weak constraint to set the boundary condition at the surface of the particle using the force terms λ_1 and λ_2 as Lagrange multipliers to enforce the no-slip condition. We then solve explicitly for λ_1 and λ_2 using second order basis functions in the expansion.

In FEMLAB a weak constraint on a boundary is given by the equation

$$0 = \int_{\partial\Omega} \mathrm{d}a\, \tilde{\nu}_i \frac{\partial R_j}{\partial u_i} \mu_j \tag{3.21}$$

with the Dirichlet condition $0 = R_i$ on $\partial\Omega$. Here ν_i is the test functions corresponding to the solution components u_i. This formulation is used to set the no-slip condition at the surface of the particle since it enables us to enter the Lagrange multipliers ourselves.

The no-slip requirement at the particle surface is a Dirichlet type boundary condition, so we choose $\boldsymbol{R} = \left[\tilde{\lambda}_1(V_1 - u_1), \tilde{\lambda}_2(V_2 - u_2), 0, \tilde{u}_1\lambda_1, \tilde{u}_2\lambda_2\right]$, where the test functions are denoted by tildes, and V_1 and V_2 are the x- and y-components of the total particle velocity (rotation and translation). Now the two Lagrange multipliers are exactly the total force needed to ensure the no-slip velocity at the surface of the particle, and Femlab allows us to choose basis functions of second order.

Using the above definitions, the equations and boundary conditions can be implemented in Femlab. In code example 2 we show how this is done.

```
% POINT CONSTRAINT TO FIX PRESSURE LEVEL
fem.pnt.shape  = {[1:3]};         % only use velocity and pressure in points
fem.pnt.ind    = {[82] [1:81]}; % in this example there are 82 points in the geometry
fem.pnt.constr = {'p' '0'};       % fix the pressure at node no 82 at the outlet

% BOUNDARY CONDITIONS: Particle, inlet, upper walls, lower walls, bumper, outlet
fem.bnd.ind    = {[Part] [Inlet] [LowerWalls] [UpperWalls] [Bump] [Outlet]};
% Bug in Femlab: flform does not preserve the order of the boundary groups
% when converting to weak form!
fem.bnd.shape = {[1:3] [1:3] [1:3] [1:3] [1:3] [1:5]};
fem.bnd.g      = {0 {'p*nx' 'p*ny' 0 0 0} {'p*nx' 'p*ny' 0 0 0} ...
                  {'p*nx' 'p*ny' 0 0 0} 0 {'nx*p' 'ny*p' 0 0 0}};
fem.bnd.r      = {{'0','0','0','0','0'} {'u_cpl-u','v_cpl-v','(p_cpl+p0)-p','0','0'} ...
                  {'u_cpl-u','v_cpl-v','p_cpl-p','0','0'} {'0','0','0','0','0'} ...
                  {'-u','-v','0','0','0'} {'0','0','0','0','0'}};
fem.bnd.weak   = {{'lm1_test*(Velx-u)' 'lm2_test*(Vely-v)' ...
                   '0' 'u_test*lm1' 'v_test*lm2'} {'0'} {'0'} {'0'} {'0'} {'0'}};
```

Femlab code 2: Point constraint and boundary conditions in Femlab notation. The spacial derivatives of the velocity components are given as the velocity component followed by the component of the direction in which the derivative is taken. The shape functions were given when the governing equations were defined.

3.3 Benchmarking the code

To check if the implementation described above is correct we will use it on some simple configurations in order to benchmark the code. For simulation of particle transport in bumper arrays the important parameters are the forces on the particle and the computation of the flow when a particle is near a solid wall. We will use the derived result for the pressure on a particle in the vicinity of a solid wall to check the reliability of the code when the particles are near the walls of the system and we will use the benchmark results published by Schäfer and Turek [24] to validate the computation of the forces on the particles.

Numerical computation of the pressure distribution on a particle near a solid wall

Because of the approximations done in the derivation, we can only expect qualitative agreement between the derived expression and the numerical result.

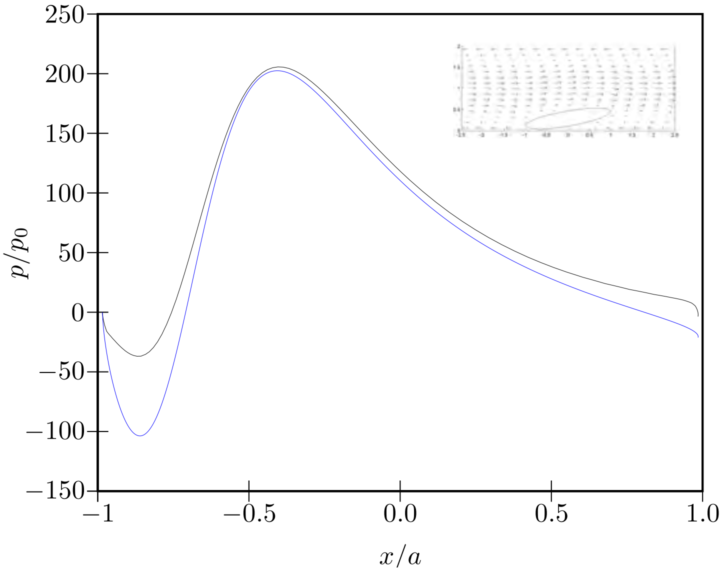


Figure 3.1: Plot of the pressure distribution on an elliptical particle near a solid wall. Comparison of numerical simulation (black line) with the analytical analysis (blue line).

In figure 3.1 we plot the pressure from a numerical simulation of a particle near a solid wall as well as the analytically derived result from section 2.3. We use the implementation described above with a slip velocity at the channel walls and a Poiseuille flow profile at the inlet. At the surface of the particle the velocity is zero which is achieved by setting V_1 and V_2 equal to 0. The pressure is normalized with $p_0 = \frac{\eta_0 U_0}{L_0}$, where $\eta_0 = 10^{-3}\,\text{Pa·s}$ is the viscosity of water and $U_0 = 10\,\mu\text{m·s}^{-1}$ is the translation velocity of the particle. The length scale used is the minimum gap width L_0 which is approximately 0.045 times the major halfaxis a of the particle.

The qualitative agreement is very good, but the magnitude is a factor two off for the negative part of the pressure.

Drag and lift forces on a cylinder

The evaluation of the forces on the particle is very important. Schäfer and Turek [24] have published benchmark results for the drag and lift on a circular cylinder in a two dimensional geometry. This is a system very similar the one we wish to solve, and they consider laminar flow. Thus we trust that our code gives valid results if it compares favorable with the published benchmarks.

The geometry for the benchmark test case is as follows: The length of the channel is 2.2 m and the channel height is 0.41 m. A circular cylinder with diameter $L_0 = 0.1\,\text{m}$ is placed slightly off center 0.2 m from the inlet and 0.2 m from the bottom wall of the channel. The flow is assumed to be a Poiseuille flow with maximum velocity $u_{\max} = 0.3\,\text{m·s}^{-1}$ in the center of the channel. The fluid has the density $\rho_0 = 1.0\,\mathrm{kg{\cdot}m^{-3}}$ and viscosity $\eta_0 = 10^{-3}\,\mathrm{Pa{\cdot}s}$. These parameter values gives a Reynolds number of $Re = \frac{\rho_0 u_{\mathrm{max}} L_0}{\eta_0} = 30$.

We nondimensionalize the equations using these values as the typical scales of the system and obtain the geometry shown in figure 3.2. The boundary conditions are no-slip velocity at the walls of the channel and at the surface of the particle. At the inlet we prescribe the Poiseuille velocity profile and at the outlet we fix the pressure to zero. The benchmark results we want to obtain is the drag coefficient defined as

$$C_{\mathrm{drag}} = \frac{2F_x}{\rho_0 u_{\mathrm{mean}}^2 L_0} \tag{3.22}$$

and the lift coefficient

$$C_{\mathrm{lift}} = \frac{2F_y}{\rho_0 u_{\mathrm{mean}}^2 L_0}, \tag{3.23}$$

where F_x and F_y are the forces in the x- and y-direction, respectively, and $u_{\mathrm{mean}} = \frac{2}{3} u_{\mathrm{max}}$ is the average velocity of the undisturbed Poiseuille flow.

We use the weak constraint and the Lagrange multipliers to evaluate the forces as described above. Because we have nondimensionalized the system we have the nondimensional quantities $\tilde{\rho} = \tilde{\eta} = \tilde{u}_{\mathrm{max}} = 1$ and the expressions for the drag coefficient and the lift coefficient becomes $C_{\mathrm{drag}} = \frac{9\tilde{F}_x}{2Re}$ and $C_{\mathrm{lift}} = \frac{9\tilde{F}_y}{2Re}$, where the nondimensional forces are the integrals of the Lagrange multipliers λ_1 and λ_2 over the surface of the cylinder, respectively.

With our setup we get the coefficients $C_{\mathrm{drag}} = 5.5791$ and $C_{\mathrm{lift}} = 0.0108$. The article requires the drag coefficient to be in the interval $[5.5700, 5.5900]$ and the lift coefficient to be in the interval $[0.0104, 0.0110]$. Thus both the drag and the lift lies in the centers of the benchmark intervals. These results makes us trust that our implementation gives the correct results in the geometries under study.

Table 3.1: Drag and lift coefficients obtained using the code developed in this thesis and corresponding intervals of accepted values from benchmark studies [24].

	C_{drag}	C_{lift}
Benchmark	$[5.5700, 5.5900]$	$[0.0104, 0.0110]$
This work	5.5791	0.0108

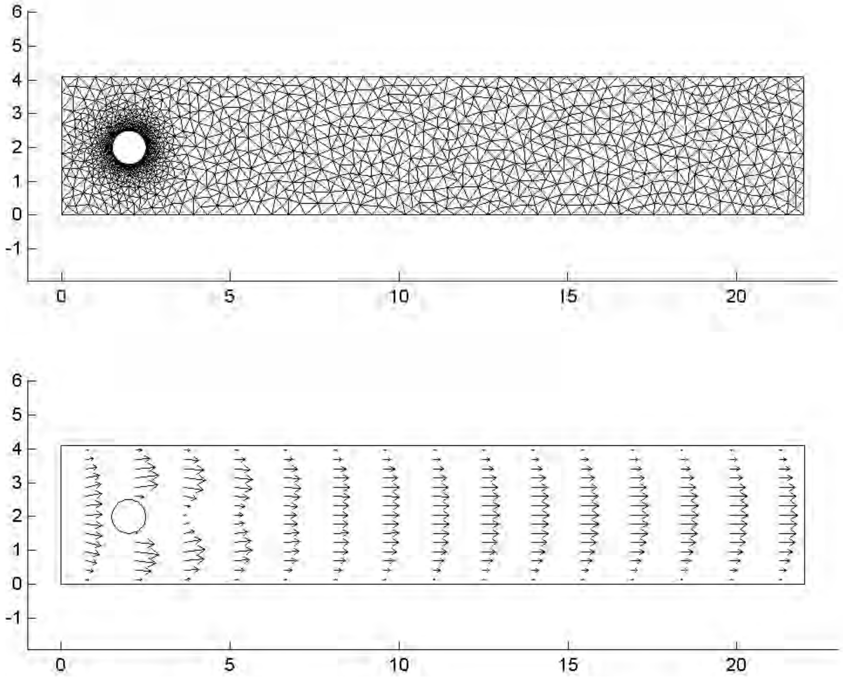


Figure 3.2: The upper figure shows the mesh used in the benchmark calculation and the lower figure shows the velocity field. The maximum mesh size is equal to the radius of the cylinder far away from the cylinder and ten times smaller at the surface of the cylinder.

Chapter 4

Flow in 2D bumper arrays

Bumper arrays are a new class of microfluidic separation devices [14]. Bumper arrays utilize the properties of the laminar flow found in the micro regime to yield deterministic separation of particles based on their size, as already mentioned in Fig. 1.2 on page 4.

Conventional separation devices rely on diffusion based separation. This type of separation relies on the correlation between average diffusion length and particle size. However the resolution of the separation in such devices is limited by the fact that diffusion is a random process. Moreover, the speed at which particles can be separated is limited by the

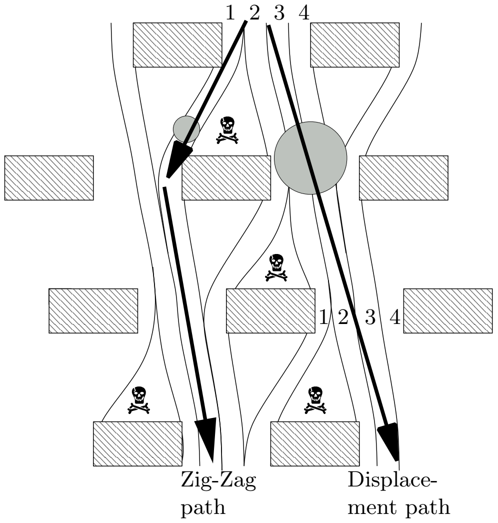


Figure 4.1: The deterministic separation in bumper arrays relies on small particles following the flow of the fluid. Particles with a radius larger than the critical radius r_c gets displaced due to interaction with the bumpers. Experiments as well as simulations show that small particles can get stuck in "dead zones" in the flow. This is regions with very low flow velocities, here marked with a skull and bones sign ☠. The number of flow lanes N depend on the periodicity of the bumper array. We number the flow lanes in the gap between two posts from 1 to N starting with the lane closest to a post on the left.

characteristic diffusion time.

In order to effectively use a bumper array it is of course necessary to know which particles that follow the zig-zag path and which that follow a displacement path. This is determined by the size of the particles. If the particles have a radius that is larger than the width of lane 1 then the particles will be pushed out into the next lane. This is called the displacement path. If the particles are sufficiently small, they are not displaced by the obstacles. Small particles are therefore passively convected by the flow. Because of the periodicity of the bumper array this is called the zig-zag path.

Since the path followed by a particle is determined by the size of the particle, there must exist a critical radius r_c defined such that particles with a radius smaller that r_c will follow the zig-zag path and particles with a radius larger than r_c will follow the displacement path. In the following we will make an estimate of the critical radius in a two dimensional bumper array.

4.1 Critical radius in two dimensions

We will derive an analytical expression for the critical radius in a two dimensional bumper array under the assumption that the flow between the bumpers can be described as a Poiseuille flow in an infinitely wide channel of the height d. This critical radius we will compare to the critical radius found through numerical simulation of a two dimensional bumper array.

In the analytical models we will use the definition of the critical radius proposed by Huang *et al.* [14]. They define the critical radius as the width of the lane nearest to the bumpers. Because we are considering flows on the micro scale, the flow is laminar. This means that each lane of the flow must carry the same amount of fluid. Therefore the flowrate Q in each lane is the same. Thus, the critical radius is found by solving

$$\frac{1}{N} = \frac{Q(r_c)}{Q_{\mathrm{total}}} \tag{4.1}$$

for r_c.

If we neglect the effect of friction on the walls the critical radius is of course just dN^{-1}. However, if we take the friction into account, the critical radius becomes larger and we must expect a correction α to the above relation

$$\frac{r_c}{d} = \alpha\frac{1}{N}. \tag{4.2}$$

The velocity of a pressure driven flow has only nonzero components in the x-direction and is described by the well known parabolic Poiseuille profile

$$u_x^{(2\mathrm{D})} = \frac{\Delta p}{2\eta L}(d-z)z, \tag{4.3}$$

where Δp is the pressure difference that drives the flow, η is the viscosity of the fluid, L is the length of the channel and d is the height of the channel.

The flow rate $Q^{(2\mathrm{D})}$ in a section of the flow with width w and height r from the lower wall can be found by integration

$$\begin{aligned} Q^{(2\mathrm{D})} &= \int_0^w \mathrm{d}y \int_0^r \mathrm{d}z\, u_x^{(2\mathrm{D})}(z) \\ &= \frac{w\Delta p}{2\eta L}\big(\tfrac{1}{2}dr^2 - \tfrac{1}{3}r^3\big). \end{aligned} \tag{4.4}$$

The fraction of the flow from 0 to r is therefore

$$\frac{Q^{(2\mathrm{D})}(r)}{Q^{(2\mathrm{D})}(d)} = 3\Big(\frac{r}{d}\Big)^2 - 2\Big(\frac{r}{d}\Big)^3. \tag{4.5}$$

If the bumper array is N periodic, each lane must carry N^{-1} of the flow. Thus the critical radius can be found from

$$\frac{Q^{(2\mathrm{D})}(r_c)}{Q^{(2\mathrm{D})}(d)} = 3\Big(\frac{r_c}{d}\Big)^2 - 2\Big(\frac{r_c}{d}\Big)^3 = \frac{1}{N}. \tag{4.6}$$

This equation can not be solved explicitly, but we might consider the solution in the limit of large N. In this limit the critical radius is much smaller than the width of the channel. Thus we can apply the approximation

$$3\Big(\frac{r_c}{d}\Big)^2 = \frac{1}{N} \qquad \text{, for } r_c \ll d. \tag{4.7}$$

Solving this equation for r_c yields

$$r_c = \sqrt{\frac{N}{3}}\frac{d}{N} = \frac{1}{\sqrt{3}}\frac{d}{\sqrt{N}}. \tag{4.8}$$

Using the approximation (4.2) we find that the correction factor is

$$\alpha = \sqrt{\frac{N}{3}} \tag{4.9}$$

in the limit of large N.

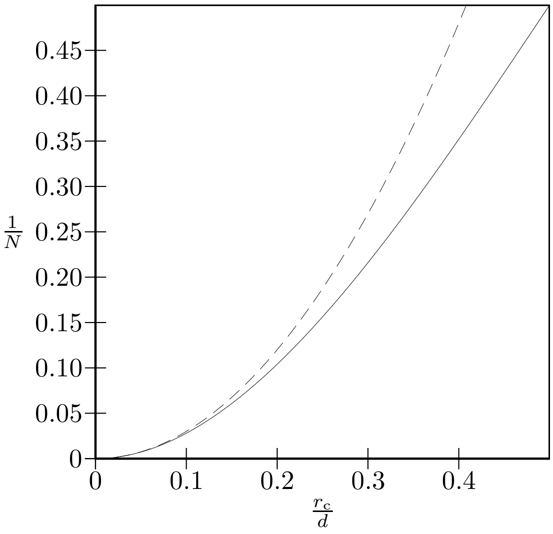


Figure 4.2: Plot of the critical radius (full line) and the approximation for large N (dashed line).

4.2 Numerical investigation of the critical radius

The analytical derivation of the critical radius builds on two assumptions

- The flow between the posts can be approximated as a Poiseuille flow in an infinitely long channel.
- The critical radius of the particles is equal to the width of the lane nearest to the first post.

Using numerical simulations of the flow in a bumper array we may verify the assumptions. We will take a bumper array produced by Jason Beech at Lund University as the basis for the geometry used in the simulations. This geometry consists of a regular array of posts with a circular cross section. The posts have a center-to-center distance $\lambda = 34\,\mu\text{m}$. The radius of the posts is $R_{\text{post}} = 8.5\,\mu\text{m}$. The array is tilted an angle $\alpha = \text{atan}(1/10)$ with respect to the direction of the channel. The width of the channel is $W = 10\lambda/\cos(\alpha)$ and the length of the channel is $L = 50W$. We assume the fluid to be water with the density $\rho = 1 \times 10^3\,\text{kg·m}^{-3}$ and viscosity $\eta = 1 \times 10^{-3}\,\text{Pa·s}$, and we choose a typical velocity for the fluid to be $u_0 = 100\,\mu\text{m·s}^{-1}$.

Using these values we find that the Reynolds number is approximately

$$Re = \frac{1 \times 10^3\,\text{kg·m}^{-3} \times 100\,\mu\text{m·s}^{-1} \times 17\,\mu\text{m}}{1 \times 10^{-3}\,\text{Pa·s}} = 1.7 \times 10^{-3}\,, \tag{4.10}$$

where we have used the width $d = 17\,\mu\text{m}$ of the gap between two neighboring posts as the typical distance for the system. This means, that we can be sure that the flow is laminar, which is required for the bumper array to work.

Because of the periodicity in the length direction, we can reduce the computational domain to a quadratic cell with side length W. The boundary conditions are no-slip at the posts and at the top and bottom wall of the geometry. The velocity is periodic from outlet to inlet and the flow is driven by a pressure difference Δp from inlet to outlet. We implement these settings as described in section 3.2 in the previous chapter.

We solve the system as a stationary problem. The solution is plotted in figure 4.4(a). We determine the critical radius using two different approaches:

1. Using streamlines, we find two streamlines ending at the top and bottom of the same post. The streamlines are initiated at the line connecting the center points of the posts on either side of the previous gap.

2. We calculate the fraction of the flow that passes the gap by integrating the velocity field along the line mentioned in item 1. The critical radius is assumed to be the distance along the line through which $\frac{1}{10}$ of the flow passes.

Using method 1, we find the critical radius to be between 0.172 and 0.175 times the gap width. The two values are due to the fact that streamlines starting between these two values end in a dead zone in front of the next post. Method 2 gives us a critical radius of 0.184 times the gap width.

Experimental values for the critical radius is $\frac{r_c}{d} \in [0.25, 0.28]$. This is considerably larger than the value of $\frac{r_c}{d} = 0.196$ predicted by the analytical model as well as the values found by numerical simulation. This suggests that a two dimensional model does not predict the size of the critical radius correctly and we need to take other effects into account, such as the nearby presence of solid walls at the top and bottom of the channel. In chapter 5 we will derive a three dimensional extension of the analytical model presented in this chapter and carry out numerical simulations of the flow in a unit cell of a bumper array with the same dimensions as the physical device produced at Lund University.

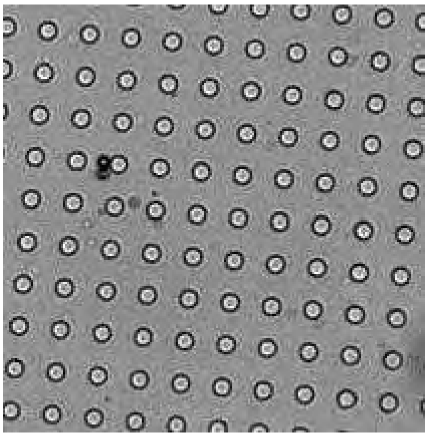


Figure 4.3: A unit section of the bumper array produced by Jason Beech at Lund University. The diameter of the posts are 17 µm. The posts are arranged on a regular quadratic grid with a center-to-center distance of 34 µm between the posts. The array of posts are tilted an angle $\alpha = \text{atan}(1/10)$ with respect to the direction of the channel.

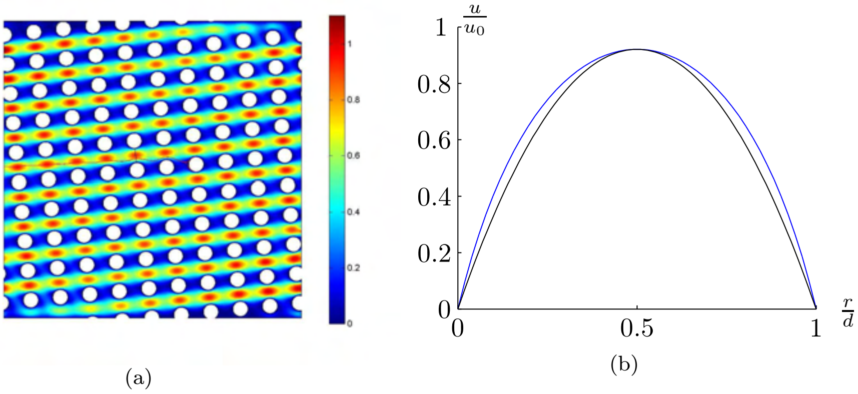
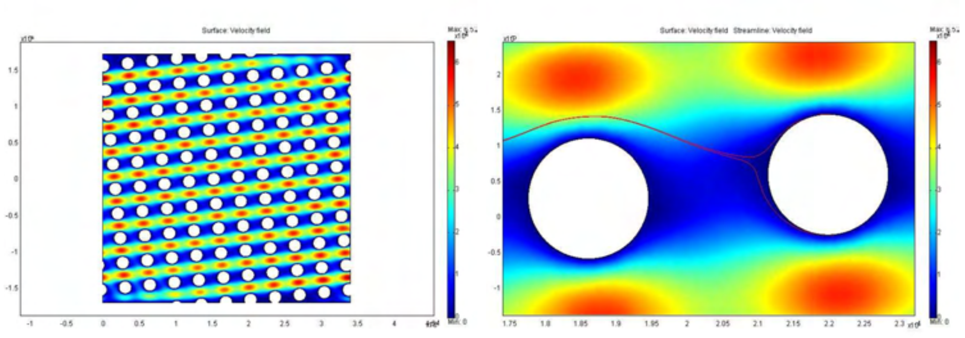


Figure 4.4: (a) Surface plot of the velocity relative to the characteristic velocity u_0. The red lines are streamlines ending at the top and bottom of a post. The critical radius is equal to the distance from the previous post to the streamlines. (b) The blue curve is the normalized velocity along the line indicated by the black line in figure (a). The black curve is the Poiseuille velocity profile with the same maximal velocity.

4.3 Dead zones

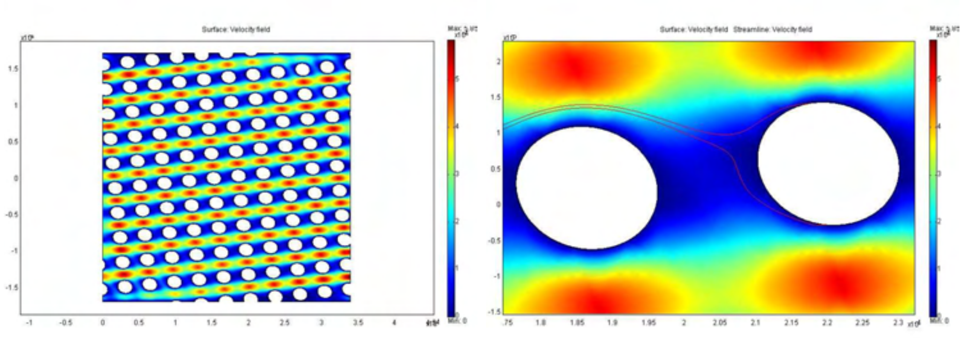
During experiments it has been observed that small particles tend to get stuck in regions, denoted dead zones, near the bumpers (see also Fig. 4.1). We may get an idea of the shape of the dead zones by studying a 2D simulation of the flow in a bumper array with the geometry shown in figure 5.5. We use periodic boundary conditions from right to left and no-slip velocity conditions on all other walls. The shape of the dead zone is determined by inspection of stream lines stopping on the top and bottom of a post. All stream lines starting in the area enclosed by these two stream lines will end at the post in stead of continuing through the hole device.



(a)

(b)

Figure 4.5: (a) The velocity field in a unit of the bumper device. The boundary conditions are periodic from left to right and no-slip velocity conditions on all other walls. (b) A closeup of two bumpers is shown together with two streamlines stopping on the top and bottom of the right bumper. The area enclosed by the two streamlines indicates the dead zone.



(a)

(b)

Figure 4.6: (a) The velocity field in a unit of the bumper device with elliptical posts. The boundary conditions are periodic from left to right and no-slip velocity conditions on all other walls. (b) A closeup of two bumpers is shown together with two streamlines stopping on the top and bottom of the right bumper. The area enclosed by the two streamlines indicates the dead zone.

Changing the shape of the bumpers changes, not surprisingly, the shape of the dead zones. This was verified by inspecting the dead zones formed in front of circular as well as elliptical posts. In figure 4.6 the velocity field and the dead zone are shown for a bumper array similar to the one shown in figure 4.5(a) but with elliptical posts. The dead zones formed in front of the elliptical posts were in this experiment much larger than the corresponding dead zones formed in front of the circular posts.

In many cases it would be desirable to optimize the devise by minimizing the dead zones in order to avoid clogging. This were the direct reason for the simulations shown here. We have not carried out further investigations of the dead zones but the approach sketched here is a capable of quickly providing design guidelines for physical devices.

One might speculate that future bumper devices could rely, in one way or an other, on the shape of the dead zones. Smaller particles tend to get stuck in the dead zones more easily than larger particles. This introduces a delay-effect that might be of interest for some applications.

Chapter 5

Flow in 3D bumper arrays

In this chapter we will expand the derived two dimensional expression for the critical radius to three dimensions and compare the result to the experimental results obtained by Jason Beech at Lund University.

5.1 Critical radius in three dimensions

In three dimensions the flow velocity is constrained to zero at the top and bottom of the channel wall as well as at the surface of the obstacles. The geometry of the narrow region between two posts is as sketched in figure 5.1

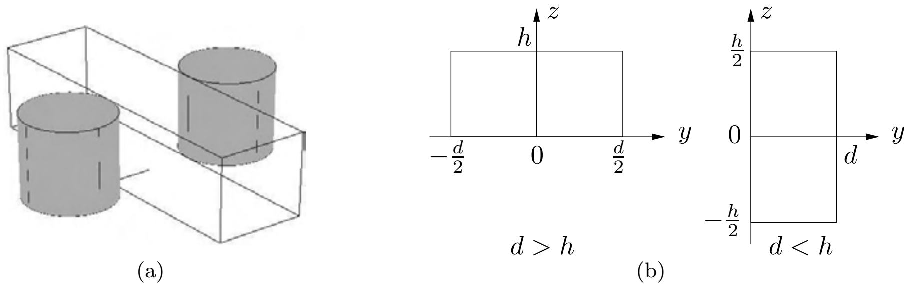


Figure 5.1: (a) The flow profile in the region between two posts is approximately the same as the flow profile of a Poiseuille flow in an infinitely long channel. (b) The coordinate systems used in the calculation of the critical radius in a three dimensional bumper array. The flow is in the x-direction out of the paper.

Again we will use the approximation that the flow between two posts can be approximated by a Poiseuille flow in an infinitely long channel. We also use the assumption that the critical radius is the width of the section of the flow nearest to a post that carries a fraction of the flow corresponding to the number of lanes.

In a channel with a rectangular cross section with width d and height h, the x-component of the velocity field is given by the infinite sum

$$u_x(y,z) = \frac{4h^2\Delta p}{\pi^3\eta L}\sum_{n,\mathrm{odd}}^{\infty}\frac{1}{n^3}\left[1-\frac{\cosh\left(n\pi\frac{y}{h}\right)}{\cosh\left(n\pi\frac{d}{2h}\right)}\right]\sin\left(n\pi\frac{z}{h}\right) \qquad \text{, for } d>h, \tag{5.1}$$

where $-\frac{d}{2} < y < \frac{d}{2}$ and $0 < z < h$. In case $h > d$ we obtain a similar result by rotation

$$u_x(y,z) = \frac{4d^2\Delta p}{\pi^3\eta L}\sum_{n,\mathrm{odd}}^{\infty}\frac{1}{n^3}\left[1-\frac{\cosh\left(n\pi\frac{z}{d}\right)}{\cosh\left(n\pi\frac{h}{2d}\right)}\right]\sin\left(n\pi\frac{y}{d}\right) \qquad \text{, for } d<h, \tag{5.2}$$

where $-\frac{h}{2} < z < \frac{h}{2}$ and $0 < y < d$. The total flow rate $Q^{(3\mathrm{D})}(d)$ is given by the the integral of the x-component of the velocity over the entire gap width

$$Q^{(3\mathrm{D})}(d) = \int_{\mathrm{gap}} \mathrm{d}a\, u_x(y,z), \tag{5.3}$$

where the velocity component $u_x(y,z)$ is given by Eq. (5.1) or Eq. (5.2) depending on the geometry of the gap.

First we will consider the case where the distance d between the posts is larger than the height h of the posts. In this case we can find the amount of liquid streaming in the section of the flow nearest to the post by integration of Eq. (5.1)

$$Q^{(3\mathrm{D})}(r) = \int_{-\frac{d}{2}}^{r-\frac{d}{2}} \mathrm{d}y \int_0^h \mathrm{d}z\, u_x(y,z). \tag{5.4}$$

Thus the flow rate in this section $Q^{(3\mathrm{D})}(r)$ of the flow relative to the total flow rate $Q^{(3\mathrm{D})}(d)$ is

$$\frac{Q^{(3\mathrm{D})}(r)}{Q^{(3\mathrm{D})}(d)} = \frac{r}{d}\frac{1-\sum\limits_{n,\mathrm{odd}}^{\infty}\frac{1}{n^5}\frac{192}{\pi^5}\frac{h}{r}\left[\tanh\left(n\pi\frac{r}{2h}\right)\cosh^2\left(n\pi\frac{r}{2h}\right)-\tanh\left(n\pi\frac{d}{2h}\right)\sinh^2\left(n\pi\frac{r}{2h}\right)\right]}{1-\sum\limits_{n,\mathrm{odd}}^{\infty}\frac{1}{n^5}\frac{192}{\pi^5}\frac{h}{d}\tanh\left(n\pi\frac{d}{2h}\right)}. \tag{5.5}$$

In case the distance between the posts is smaller than the height of the posts, the relative flow rate is found from the integral

$$Q^{(3\mathrm{D})}(r) = 2\int_0^r \mathrm{d}y \int_0^{\frac{h}{2}} \mathrm{d}z\, u_x^{(3\mathrm{D})}(y,z). \tag{5.6}$$

The relative flow rate is therefore

$$\frac{Q^{(3\mathrm{D})}(r)}{Q^{(3\mathrm{D})}(d)} = \frac{1}{2}-\frac{1}{2}\frac{\frac{96}{\pi^4}\sum\limits_{n,\mathrm{odd}}^{\infty}\frac{1}{n^4}\cos\left(n\pi\frac{r}{d}\right)-\sum\limits_{n,\mathrm{odd}}^{\infty}\frac{1}{n^5}\frac{192}{\pi^5}\frac{d}{h}\tanh\left(n\pi\frac{h}{2d}\right)\cos\left(n\pi\frac{r}{d}\right)}{\sum\limits_{n,\mathrm{odd}}^{\infty}\frac{1}{n^5}\frac{192}{\pi^5}\frac{d}{h}\tanh\left(n\pi\frac{h}{2d}\right)}. \tag{5.7}$$

Figure 5.2 shows a series of plots of $\frac{1}{N}$ as a function of $\frac{r}{d}$ for channels with different height to width ratios. The maximum particle size is always limited by the shortest dimension of the gap. Thus for $h > d$ the curves are very close together, and the maximum particle radius is half the gap width. For $h < d$ the height of the gap limits the maximum size of particles that can pass through the bumper array.

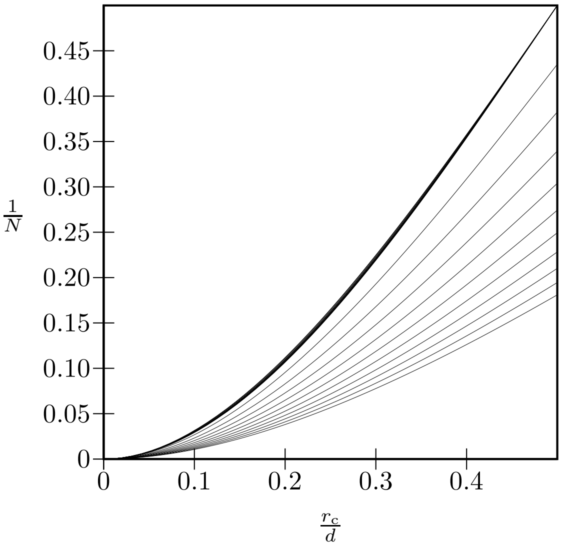


Figure 5.2: The flow rate $Q^{(3D)}(r)$ relative to the total flow rate $Q^{(3D)}(d)$ as a function of the position r between the posts relative to the minimum of the gap width d or the height h as a function of the periodicity N of the bumper array. The graphs represent the relative flow rate in bumper geometries with height to width ratios ranging from $\frac{1}{2}$ to 2.

5.2 Simulation of the flow in a three dimensional bumper array

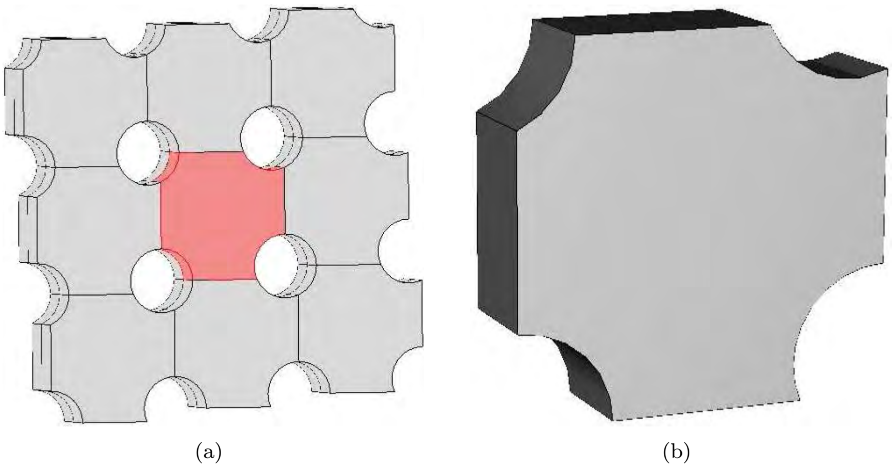
In order to simulate the flow in a three dimensional bumper array the computational domain must be reduced even more than the domain used in the two dimensional simulation described in chapter 4. We will neglect the no-slip conditions at the outer walls of the channel and reduce the computational domain to a unit cell in the middle of the channel. Assuming that the flow in this cell is identical to the flow in the neighboring cells we can use periodic boundary conditions. We can reduce the size of the unit cell even more by realizing that the flow must be symmetric around the center plan of the channel (see figure 5.3).

We use boundary conditions as described in section 3.2 for the periodic and the no-slip boundaries. At the symmetry plane the viscous stress is zero and the velocity component perpendicular to the symmetry plane is zero. The flow is driven by a pressure difference from the inlet to the outlet. With this setup, we obtain the solution shown in figure 5.4.

Using the definition of the critical radius as suggested by Huang *et al.* [14] we find by carrying out the integration

$$\frac{\int_0^{r_c} \mathrm{d}a\, \boldsymbol{u}\cdot\boldsymbol{n}}{\int_{\mathrm{inlet}} \mathrm{d}a\, \boldsymbol{u}\cdot\boldsymbol{n}} = \frac{1}{N} \tag{5.8}$$

that the critical radius is $r_c/d = 0.180$. This result can be controlled by realizing that the flow through the bottom boundary must stem from the fluid contained in the section of the inlet flow that is within a distance of the critical radius from the bottom post. Indeed the



(a) (b)

Figure 5.3: The geometry of the three dimensional bumper array is assumed to be periodic and symmetric about the center plan of the array. Thus a periodic geometry as shown in figure (a) can be reduced to the unit cell shown in (b) where the periodicity of the total array is modeled using periodic inlet and outlet conditions and a symmetry condition at the center plan of the geometry.

same result is obtained if we solve the equation

$$\frac{\int_0^{r_\mathrm{c}} \mathrm{d}a\, \boldsymbol{u}\cdot\boldsymbol{n}}{\int_{\mathrm{bottom}} \mathrm{d}a\, \boldsymbol{u}\cdot\boldsymbol{n}} = 1, \tag{5.9}$$

where $\boldsymbol{n}$ is a normal vector to the surfaces.

Similarly to the approach used in the two dimensional investigation we can use streamlines to find the critical radius. Using this method we find that the critical radius is a function of the position in the bumper array. In table 5.1 the critical radius at different positions in the bumper array is tabulated.

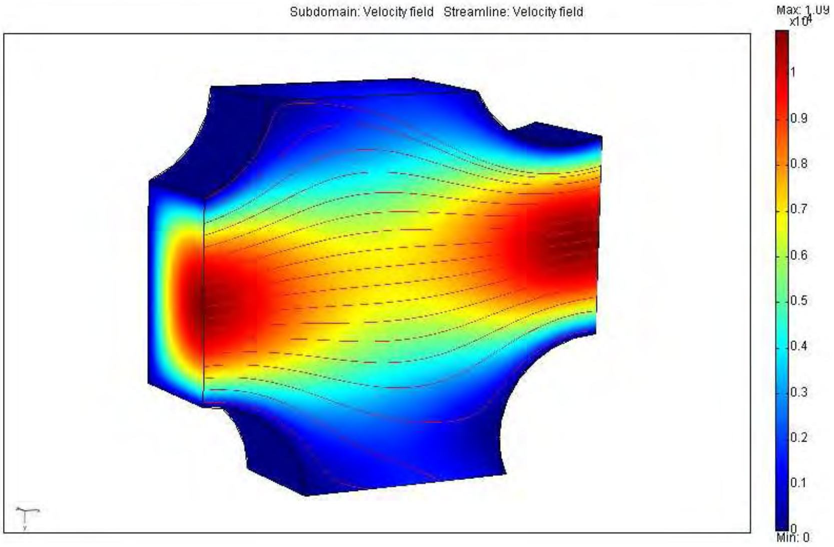


FIGURE 5.4: The velocity field in half a unit cell of the bumper array used by Jason Beech. The boundary conditions are periodic from inlet to outlet and from top to bottom. No-slip velocity conditions are applied on the posts and on the walls of the channel. At the center plan of the unit cell a symmetry boundary condition is applied.

TABLE 5.1: Corresponding values of position in the bumper array and critical radius determined using streamlines. The relative position $\frac{2z}{h}$ is given with respect to the center of the channel.

$\frac{2z}{h}$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\frac{r_c}{d}$	0.1431	0.1442	0.1448	0.1460	0.1485	0.1512	0.1561	0.1626	0.1719	0.1838

5.3 Experimental results

At Lund University, under supervision of Jonas Tegenfeldt, Jason Beech has carried out experiments with bumper arrays fabricated in the polymer PDMS. The geometry of the bumper array was as shown in figure 5.5. The dimensions are as mentioned in the caption of the figure. The total length of the channel containing posts is 50 unit cells and the total width is one unit cell. Experimentally the value of the critical radius relative to the gap width is found to be within the interval $[0.25, 0.28]$.

Huang *et al.* [14] reports results from a bumper array with gap width $d_{\text{Huang}} = 1.6\,\mu\text{m}$. The periodicity of the array is $N_{\text{Huang}} = 10$ similarly to the array used by Beech. The observed critical diameter in Huang's device is approximately $0.8\,\mu\text{m}$. This means that the critical radius relative to the gap width is $\frac{0.8\,\mu\text{m}}{2\times1.6\,\mu\text{m}} = 0.25$.

In table 5.2 we summarize the results for the critical radius obtained from the numerical investigations as well as the experimentally observed values. The agreement between the

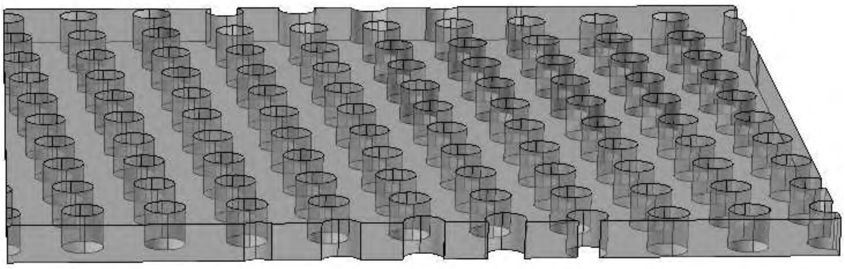


Figure 5.5: Unit cell of a bumper array. The height of the channel is $h = 20\,\mu\text{m}$, the distance between the posts is $d = 17\,\mu\text{m}$ and the center to center distance between the posts is $34\,\mu\text{m}$. The bumper array is periodic with period $N = 10$.

Table 5.2: Comparison of the observed critical radius with the simulated values.

	Tool	$\frac{r_c}{d}$	Method
3D	Observed	0.25–0.28	Experiments at Lund University
	Observed	0.25	Experiments by Huang *et al.*
	Simulation	0.143–0.184	Investigation of streamlines
	Simulation	0.180	Flow rate calculation
	Theory	0.190	Assuming Poiseuille flow
2D	Simulation	0.172–0.175	Investigation of streamlines
	Simulation	0.184	Flow rate calculation
	Theory	0.196	Assuming Poiseuille flow

experimental values are good. However the numerically obtained values are approximately 25 % lower than the experimental data. This implies that the assumption that the critical radius is simply the width of the first lane is not correct. At the same time it underlines the fact that the finite size of the particles must be taken into account if we wish to study the working principles of bumper devices numerically.

It seems counterintuitive that the experimentally observed critical radii are larger than the ones predicted by the simple flow analysis. This suggests some kind of “opposite lift-effect”implying that the actual geometry of the narrow region between the posts, and the flow pattern in the region between two consecutive rows of posts, are more important than accounted for in the laminar flow model with equal splitting of the flow between the lanes in the bumper array.

In chapter 6 we will investigate the motion of finite-sized particles in bumper arrays through numerical simulations of the motion of particles based on the implementation described in chapter 3.

Chapter 6

Particles in a bumper array

This chapter will be devoted to direct numerical simulations of finite-sized particles in bumper arrays. We will present simulations done on circular as well as elliptical particles in a bumper array with circular posts. Pori to the simulations presented here, the numerical code were used on several test cases in order to verify the output from the simulations. We will not present these test cases here. In stead we will focus on the simulations of bumper arrays, as this has been the major point of interest for the project. However the test studies gave us some insight in the behavior of particles in different channels geometries and based on this knowledge we have made qualified guesses on the initial orientations of the non-spherical particles as we will discuss later.

6.1 Time stepping and mesh generation

As mentioned in chapter 3 we have implemented the time-stepping as a MATLAB loop of solutions to the time-independent Navier-Stokes equation for each new position of the particle. We have implemented two time stepping algorithms; a forward Euler time step and a Runge-Kutta time step.

Euler times step

The output from each solution to the flow problem in a given geometry is the state of the liquid given by the velocity $\boldsymbol{u}$ and the pressure p and the forces on the particle. Using these we can solve Eq. (2.26) and Eq. (2.30). For the Euler time step Δt_i, we simply use the discretizations

$$\boldsymbol{v}_{i+1} = \boldsymbol{v}_i + \frac{1}{m}\boldsymbol{F}_i \Delta t_i \tag{6.1}$$

and

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \boldsymbol{v}_i \Delta t_i + \frac{1}{2m}\boldsymbol{F}_i \Delta t_i^2 \tag{6.2}$$

for the position $\boldsymbol{r}_i$ and velocity $\boldsymbol{v}_i$ of the particle in the i'th time step, where $m = \int_P \mathrm{d}\,V\,\rho_P$ is the mass of the particle. $\boldsymbol{F}_i$ is the total force working on the particle, and the discretization

$$\boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_i + \frac{1}{I}\boldsymbol{\tau}_i \Delta t_i \tag{6.3}$$

and

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \boldsymbol{\omega}\Delta t_i + \frac{1}{2I}\boldsymbol{\tau}_i \Delta t_i^2 \tag{6.4}$$

is used for the orientation $\boldsymbol{\theta}$ and angular velocity $\boldsymbol{\omega}$ of the particle. I is the moment of inertia and $\boldsymbol{\tau}$ is the torque about the axis of rotation through the center of mass of the particle.

Runge-Kutta time step

The Runge-Kutta time step were implemented as the classical fourth order Runge-Kutta time step where

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \frac{\Delta t}{6}\Big(k_1 + 2k_2 + 2k_3 + k_4\Big), \quad t_{i+1} = t_i + \Delta t, \tag{6.5}$$

where the coefficients $k_j, j = 1, 2, 3, 4$ are given as

$$k_1 = \boldsymbol{v}\big(\boldsymbol{r}_i, t\big), \qquad \mathcal{R}_1 = \boldsymbol{r}_i + \tfrac{1}{2}\Delta t k_1, \tag{6.6a}$$

$$k_2 = \boldsymbol{v}\big(\mathcal{R}_1, t + \tfrac{1}{2}\Delta t\big), \qquad \mathcal{R}_2 = \boldsymbol{r}_i + \tfrac{1}{2}\Delta t k_2, \tag{6.6b}$$

$$k_3 = \boldsymbol{v}\big(\mathcal{R}_2, t + \tfrac{1}{2}\Delta t\big), \qquad \mathcal{R}_3 = \boldsymbol{r}_i + \Delta t k_3, \tag{6.6c}$$

$$k_4 = \boldsymbol{v}\big(\mathcal{R}_3, t + \Delta t\big). \tag{6.6d}$$

$\mathcal{R}_j$ is a temporary position. The orientation were calculated using an equivalent approach. We did only use the Runge-Kutta time step on some of the test cases since it requires four solutions per time step. The improved accuracy of the Runge-Kutta time step did not allow us to make much larger time steps because we need to interpolate the solution from the previous time step to the mesh in the geometry of the current geometry in order to use it as initial condition. If the particle moves more than one mesh cell, this introduces numerical errors. For the same reason we used an adaptive time step based on the translation velocity of the particle

$$\Delta t_{i+1} = \frac{\min(h_{\text{mesh}})|_{\partial P}}{|\boldsymbol{v}_i|}, \tag{6.7}$$

where h_{mesh} is the lengths of the meshelements at the surface of the particle.

Generation of mesh

As was demonstrated in the benchmark calculation in section 3.3, we obtain accurate results for the forces on the surface of a round particle if we resolve the surface with a mesh with elements one tenth of the radius of the particle. We use this length as the maximum mesh size at the particle surface. If the particle is near the bumpers we need to resolve the flow in this region. To do so we generate at least three mesh elements in narrow regions of the geometry.

The channel geometry stays the same throughout all simulations, but the geometry of the computational domain changes because the particle is represented by a boundary of the computational domain. Therefore it is necessary to generate a new mesh every time step.

6.2 Modeling of the particle-wall contact force

In order to take the interaction between the solid particles and the walls of the microsystems into account we need to model the contact force $\boldsymbol{F}_{\text{coll}}(\boldsymbol{r}) = F_0\tilde{F}(\boldsymbol{r})\boldsymbol{e}_F(\boldsymbol{r})$ between the wall and the particle. F_0 is the physical characteristic scale of the force. As mentioned we have chosen to model the force as an extra force per area acting on the surface of the particle. The direction $\boldsymbol{e}_F$ of the force is perpendicular to the surface of the bumpers and works in a small region of width ϵ from the bumper surface.

The magnitude $\tilde{F}$ of the force is a function of the distance d_{B} from the bumper surface ∂B. The distance is simply given by

$$d_{\text{B}}(\boldsymbol{r}) = \min(|\boldsymbol{r} - \boldsymbol{r}_{\text{B}}|) \tag{6.8}$$

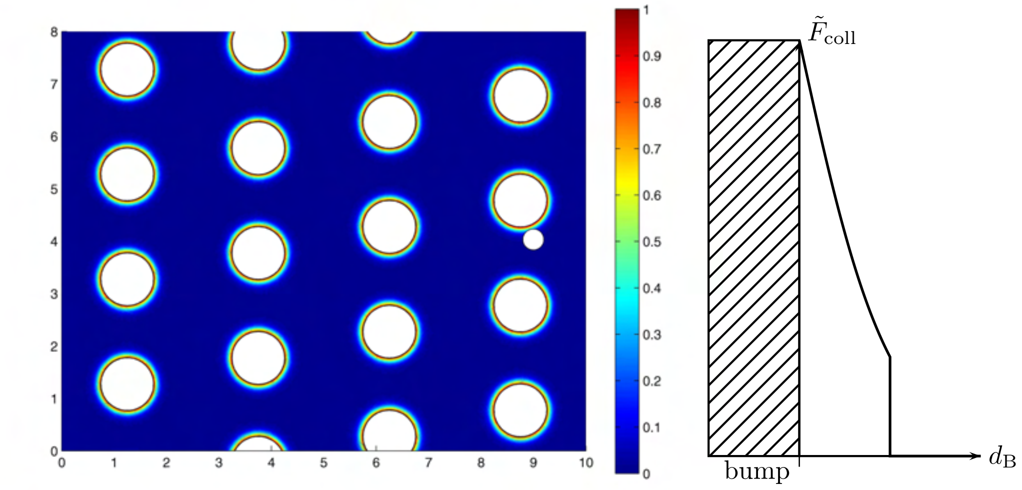
where $\boldsymbol{r}_{\text{B}} = \{\boldsymbol{r}|\boldsymbol{r} \in \partial B\}$ is a given point on the surface of the posts. The direction $\boldsymbol{e}_F$ of the force is

$$\boldsymbol{e}_F \equiv \frac{\boldsymbol{\nabla} d_{\text{B}}}{|\boldsymbol{\nabla} d_{\text{B}}|}. \tag{6.9}$$

We tried several functions as the model of the magnitude of the force. Several polynomial functions were tried as well as different exponential functions. If the magnitude were too small, the particles were not sufficiently repelled by the bumpers and penetrated into the walls. This breaks the numerical code. If, on the other hand, the magnitude of the force were too large, the particles felt a strong repulsion. This resulted in particles getting shoot away from the posts. After many numerical experiments we found that the function defined as

$$\tilde{F}_{\text{coll}} \equiv \begin{cases} 1 - \tanh\left(\frac{d_{\text{B}}}{\epsilon}\right), & d_{\text{B}} < \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{6.10}$$

worked well. A surface plot of the magnitude of the force is shown in figure 6.1(a).



(a) (b)

Figure 6.1: (a) Surface plot of the magnitude of the contact force from the bumpers. (b) Plot of $\tilde{F}_{\mathrm{coll}}$.

6.3 Circular particles

In this section we present simulations done on the dynamics of finite-sized circular particles in a bumper array. Due to numerical limitations it is not possible to simulate the flow and motion of particles in full devices. As discussed earlier, direct modeling of finite-sized particles contained in a system of comparable dimensions rely on accurate computation of the forces exerted by the fluid. To obtain this, we need to resolve the surface of the particle using a fine grid.

The critical radius is approximately the gap width divided by the periodicity of the bumper array. Thus we need to use mesh elements much smaller than the gap width, in the neighborhood of the particle if we use arrays with a large periodicity. On the other hand in bumper arrays with a small period, the critical size of particles is very close to the the gap size which also imposes problems.

As a compromise we have chosen to model a bumper array with $N = 4$ in our proof-of-concept simulations. Based on our analytical model, the critical radius in such a device is $r_{\mathrm{c}} \approx 0.33d$. The finite size of the particles affects the flow. Therefore we can not reduce the size of the computational domain to the smallest unit cell as we did in the simulation of the flow in the three dimensional bumper array. Instead we will use a unit cell containing a full period of the bumper array and use periodic boundary conditions as we did in the flow simulations in chapter 4. However, we will use periodic boundary conditions in all directions thus modeling a section of the flow in the middle of an infinitely large array.

We will use the geometry shown in figure 6.2. We use the same geometry for all simulations, varying only the size and shape of the particles. The submersing fluid is assumed to be water. We use the gap width d to nondimensionalize lengths and we will use a velocity of $100\,\mathrm{\mu m{\cdot}s^{-1}}$ as the typical velocity of the system.

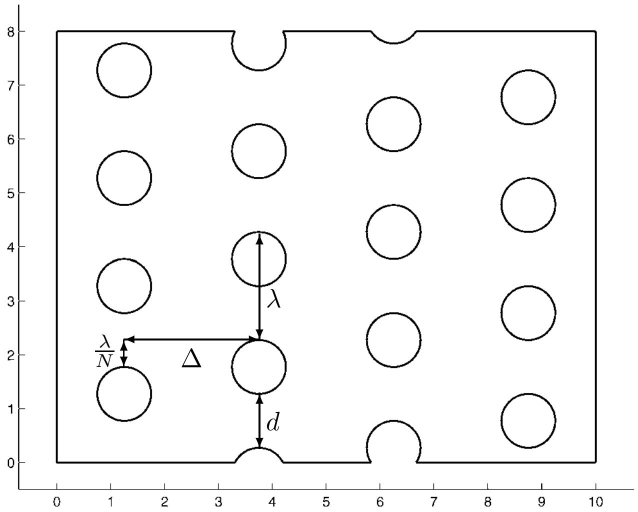


Figure 6.2: The geometry of the bumper array used in the simulations of the motion of finite-sized particles.

The typical scales used to nondimensionalize the equations as well as the geometrical properties of the bumper array are tabulated in table 6.1.

The particle is assumed to be rigid and to have the same density $\rho_P = \rho_0$ as the liquid. Many biological particles have approximately the same density as water and we are not interested in studying buoyancy effects in this simulation. Thus this is a natural choice for the density of the particle.

All particles are started with their center of mass at the same initial position. The initial velocity is set to the velocity of the flow in a bumper array with no particles at the position of the center of mass of the particle.

We initiate the particles such that we expect them to be carried by the flow and hit a bumper in the third row in the bumper array. In this way, the particles are allowed to be convected by the flow a sufficiently long way for the effects of the initial flow not

Table 6.1: The parameter values used in the simulations of the particle motion in the bumper arrays.

Density of liquid	ρ_0	$1 \times 10^3\,\mathrm{kg{\cdot}m^{-3}}$
Viscosity of liquid	η_0	$1 \times 10^{-3}\,\mathrm{Pa{\cdot}s}$
Typical velocity	u_0	$100\,\mathrm{\mu m{\cdot}s^{-1}}$
Gap width (Length scale)	d	$17\,\mathrm{\mu m}$
Bumper-bumper distance	λ	$2.0d$
Distance between rows	Δ	$2.5d$
Bumpers per period	N	4
Collision force scale	F_0	$\frac{\eta_0 u_0}{d}$
Range of force	ϵ	0.1

being perfectly no-slip at the particle surface. A particle is then considered "bumped" if it continues above the following post in the next row. If it passes under the following post in the next row it is following the zig-zag path.

The displacement path

Large particles bumping into the posts follow the displacement path as discussed in the introductional chapter of the thesis. These particles are expected to follow the same path relative to the bumpers they are in contact with after the first bump. To verify this, we examine the path of a bumped particle in the vicinity of the two last encounters with posts. These paths are plotted in figure 6.3.

The paths relative to the bumpers are very similar. This also tells us that diffusion effects can indeed be neglected because the particle paths get "reset" at post-particle contacts. Thus, the maximum time interval a large particle can diffuse is the time it takes for convection to transport the particle from one post to the next (approximate distance Δ). In this bumper array the possible diffusion time interval is therefore $\Delta/u_0 = 2.5d/u_0 = 0.425\,\mathrm{s}$. The diffusion constant for a particle is of the order of $D = 10^{-11}\,\mathrm{m}^2{\cdot}\mathrm{s}^{-1}$ giving a diffusion distance of approximately 10 % of the gap width. However, since the path of the particle gets reset at encounters with the posts this will be the maximum diffused distance independent of the total length of the bumper array.

The zig-zag path

When small particles are in free flow between the posts they are expected to follow the streamlines of the undisturbed flow, approximately. To check if this is the case, we plotted the path of a small particle as well as a flow line of an undisturbed flow starting in the same point as the particle, see figure 6.4. As is evident the particle path follows streamline of the undisturbed flow nicely. When the particle comes in the vicinity of the post in the third row of the bumper array it diverges from the streamline, as expected. The above observations makes us trust that the computed particle paths are correct.

The critical radius

We ran a series of simulations with particles of varying radii. The radii relative to the gap width were 0.2, 0.25, 0.27, 0.29, 0.30, 0.31, 0.33, 0.35, 36.5, 39.5 and 0.40. With these radii, all particles with a radius less than or equal to 0.365 followed the zig-zag path. Only the particles with radii 0.395 and 0.40 got displaced, see figure 6.5. This means that the critical radius determined from this simulations is in the interval $r_c/d \in [0.365, 0.395]$. This is between 10 % and 20 % more than what were predicted from the analysis of flow rates and streamlines presented in chapter 4.

From the simulations it is seen, that the collision forces repells the particles from the posts as intended. There is a small space between the particles and the bumpers. The width of this space is approximately $0.05d$, which must be considered satisfactory since particles and bumpers are not allowed to overlap in the implementation used in this thesis. With the length scales used here, this distance is less than one micrometer. Thus, the gap could be considered as an effect caused by surface properties such as roughness, which will always be present in real physical devices.

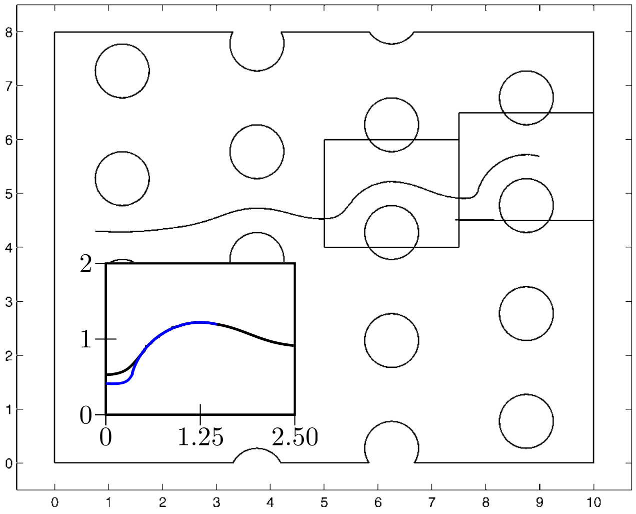


FIGURE 6.3: The path for a particle with radius $R_\mathrm{P} = 0.4d$. The two last sections of the path enclosed in rectangles are plotted on top of each other in the inset. The black curve is the path from the first rectangle and the blue curve is the part of the path enclosed in the second rectangle. The two paths fit nicely on top of each other. This indicates that when a large particle gets displaced, it will do so throughout the rest of the bumper array.

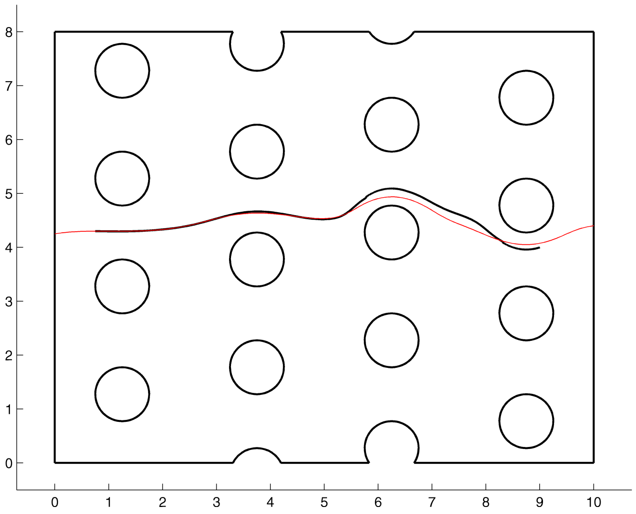


FIGURE 6.4: The path of a small particle with radius $R_\mathrm{P} = 0.25d$ (black curve) compared with a streamline of the undisturbed flow in the bumper array starting in the same point as the particle (red curve). The path of the particle and the streamline is nearly undistinguishable until the particle comes in to the vicinity of a post.

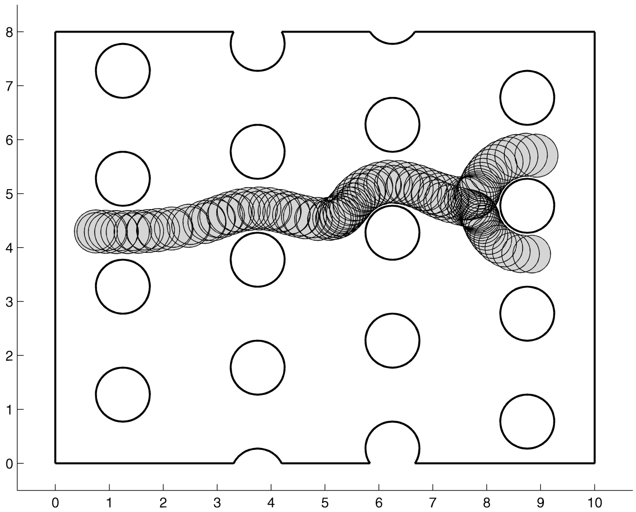


Figure 6.5: Separation of circular particles in a bumper array. The figure shows the particles with radii $R_\mathrm{P}/d = 0.35$ and $R_\mathrm{P}/d = 0.40$ at different times while they are convected through the bumper array. When the fourth row of posts are reached, the larger particle gets displaced while the smaller follows the zig-zag path.

The observed value of the critical radius in a bumper array with $N = 10$ is $33\,\%$ larger than the critical radius found from investigation of flow lines or flow-calculations as discussed in chapter 5. This makes us conclude that the critical radius can not be found from simple investigations of the flow because the finite size and the precise geometry of the bumper array plays an important role for the size of the critical radius in a specific device.

6.4 Elliptical particles

In the previous sections we verified that the developed code gives results which is in agreement with what should be expected. One of the goals for this project was to develop an implementation that could handle finite-sized particles of an arbitrary shape. To verify, that our implementation can handle noncircular particles we have made a series of simulations with elliptical particles with different eccentricities.

Handling of arbitrarily shaped particles is important if we wish to investigate the possibility of separation based on the morphology of the particles in stead of their size. In the simulations we have used particles with the same masses as the particles used in the simulations of the circular particles but we have varied the eccentricity e defined as

$$e = \sqrt{1 - \tfrac{b^2}{a^2}}, \tag{6.11}$$

where a and b are the semi axis of the ellipses. We have done simulations with particles with the following eccentricities: 0.1, 0.3, 0.5, 0.7 and 0.9. The size of the particles were as mentioned determined by the mass of the particles.

The initial position and velocity of the particles were the same as the initial position and velocity of the circular particles. The initial orientation were chosen to be with the major semi axis in horizontal direction. This was chosen because initial test studies showed us that elliptical particles tend to align themselves with the flow.

With this initial configuration we obtained particle paths as depicted in figure 6.6. Interestingly, the separation pattern were the same as for the circular particles: Particles with masses corresponding to circular particles with radii smaller than $R_\mathrm{P}/d \leq 0.365$ followed the displacement path while particles with masses corresponding to circular particles with radii larger than $R_\mathrm{P}/d \geq 0.395$ got displaced. This were independent of the eccentricity of the particles.

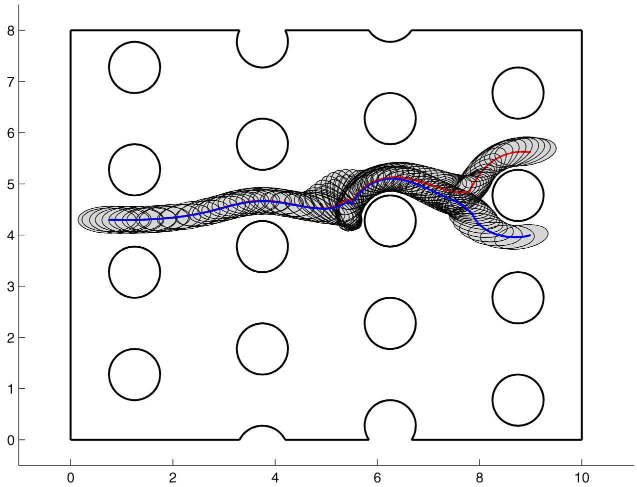


Figure 6.6: Separation of elliptical particles in a bumper array. The figure shows the particles with eccentricity $e = 0.9$ and mass similar to particles of radii $R_\mathrm{P}/d = 0.35$ and $R_\mathrm{P}/d = 0.40$ at different times while they are convected through the bumper array. When the forth row of posts are reached, the larger particle gets displaced while the smaller follows the zig-zag path. The red and blue curve are the centers of mass for the large and the small particle, respectively.

6.5 Summary of bumper results

We have carried out simulations of particle transport in bumper arrays. We focused on one particular bumper geometry with four rows per period and circular bumpers. We also carried out simulations of particle transport in other array geometries but the time did not allow us to do a systematic examination of the effect of these geometry differences. In figure 6.7 we show an example of an elliptical particle in a bumper array with rectangular posts.

The separation works independent of the geometry of the device but in some geometries the particles got stuck in the dead zones more often than in other geometries. This was especially the case when rectangular posts were used, especially if the row-to-row distance was small. Here the dead zones in front of the bumpers are much larger than in the case of circular posts.

Even in the fixed geometry chosen for our studies the parameter space is huge. We limited ourselves to studying the path of non-buoyant particles with one initial position and orientation. We carried out simulations for a series of eleven particle radii and five different eccentricities for elliptical particles with masses similar to the masses of the circular particles. The chosen radii were chosen based on the estimates for the critical radius made from investigations of the flow in bumper arrays. It turned out that the critical radius for particles of finite size was much larger than what was predicted by the initial flow studies. The critical radius fell in the top region of our interval of chosen radii where we did not have many values. The critical radius were found to be between $r_c/d = 0.365$ and $r_c/d = 0.395$ for the circular particles. Elliptical particles were separated similar to circular particles of the same mass. This was independent of the eccentricities of the particles for the parameter values used in our studies.

Our simulations were limited by long computation times (approximately two days per run). However, a more severe limitation was a memory leak in FEMLAB that made the computation stop with a "Out of memory" warning. Closing and opening of the program cleaned out the memory and the computations could continue form the last saved solution. Another problem we had during the development of our code was a bug in the way FEMLAB converts boundary expressions to weak form: The order of the boundary expressions and the corresponding shape functions is not preserved. We worked around this problem by rearranging the terms ourselves but it took us some time to locate the origin of the strange results produced from code not taking this into account.

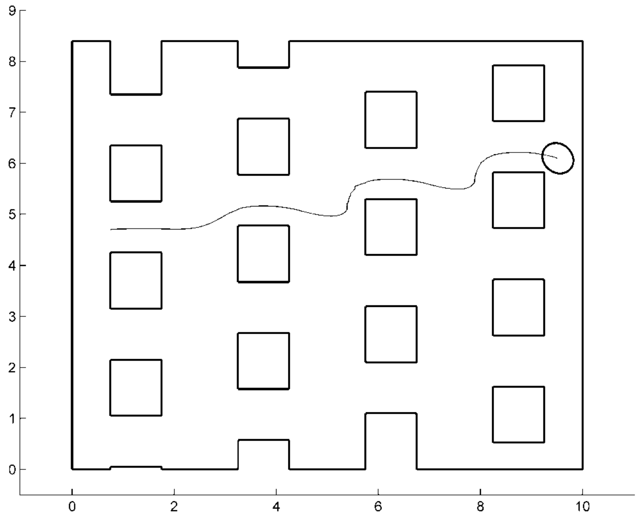


FIGURE 6.7: The path of elliptical particle in a bumper array with rectangular posts.

Chapter 7

The level set method

The direct simulation approach described in the previous chapters works and gives trustworthy results. However, the computational time is large and the geometry changes and interpolation of solutions from one mesh to a new mesh in the next time step is error-prone. It would be desirable to have a fixed geometry and mesh, and to move the particle relative to this mesh without the need to generate new meshes every time step.

In this chapter we introduce the level set method as a method for achieving this goal. We sketch the basic ideas and advantages of the level set method and derive a level set formulation of an incompressible two fluid system. We use this formulation to model the particle transport in a microsystem by representing the particle as a highly viscous liquid droplet.

7.1 The idea

The idea behind the level set method is to represent an interface by the zero level set

$$\Gamma = \{\boldsymbol{r}|\phi(\boldsymbol{r}) = 0\} \tag{7.1}$$

of some smooth function $\phi(\boldsymbol{r})$. By evolving the level set function in time we can capture the front implicitly as the zero contour of the level set function rather than following it explicitly. The time-evolution of the front is determined by the physics of the modeled problem. This formulation has a number of advantages such as

- It is completely general and thus extendable to higher dimensions and is usable for a vast range of problems concerning moving interfaces.

- Topological changes of the front (such as merging of bubbles) or collisions with hard walls are handled automatically.

- The level set function can be discretized on the same grid as the governing physical equations. Thus the same numerical tools can be used to solve for the evolution of the level set function and for the physical system.

We which to model the transport of finite-sized particles in confined geometries. In the level set formulation introduced here this is achieved by having a fixed channel geometry

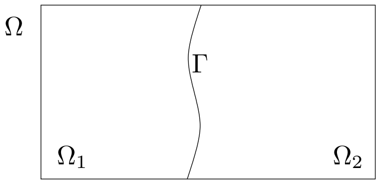


Figure 7.1: A domain Ω consisting of two subdomains Ω_1 and Ω_2. The boundary Γ between the two subdomains is the zero level set.

and a fluid with a varying viscosity. The particles are then represented by a high viscous part of the fluid and the submersing fluid is represented by the part of the fluid with low viscosity. The high viscous particle is separated from the liquid with low viscosity by the zero contour of the level set function (the zero level set).

In many biological applications, the particles are not completely rigid. Many cells, for example, consist of a flexible lipid membrane containing a soft interior. Particles represented as viscous droplets might be a useful model for cell handling in microfluidic systems.

7.2 Governing equations

In the following we derive the level set formulation for an incompressible two liquid flow in a microfluidic system. We follow the approach used by Chang et al. [6].

Consider a domain Ω consisting of two subdomains Ω_1 and Ω_2. The common boundary between Ω_1 and Ω_2 is the interface Γ which we want to evolve.

The rate of change $\frac{\partial \boldsymbol{p}}{\partial t}$ of the momentum of the fluid is given by

$$\frac{\partial \boldsymbol{p}}{\partial t} = \int_\Omega \mathrm{d}\boldsymbol{r}\, \rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} \tag{7.2}$$

where $\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t}$ is the substantial time derivative. According to Newton's second law the change in momentum must be balanced by the forces acting on the volume of fluid. In a microfluidic system we can neglect gravity, and the only force $\boldsymbol{F}_\sigma$ acting on a volume of fluid stems from the stresses σ exerted by the surrounding liquid

$$\boldsymbol{F}_\sigma = \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \mathrm{d}\boldsymbol{a}, \tag{7.3}$$

where $\boldsymbol{\sigma}$ is the stress tensor defined by

$$\sigma_{ij} = -p\delta_{ij} + \eta\left(\partial_j u_i + \partial_i u_j\right). \tag{7.4}$$

Newton's second law therefore takes the form

$$\int_\Omega \rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t}\, \mathrm{d}\boldsymbol{r} = \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \mathrm{d}\boldsymbol{a}. \tag{7.5}$$

The right hand side of this equation can be split up in three integrals: two parts for each of the boundaries of the two subdomains and one along the common interface

$$\int_\Omega \rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t}\, \mathrm{d}\boldsymbol{r} = \int_{\partial\Omega_1} \boldsymbol{\sigma} \cdot \mathrm{d}\boldsymbol{a} + \int_{\partial\Omega_2} \boldsymbol{\sigma} \cdot \mathrm{d}\boldsymbol{a} + \int_\Gamma [\boldsymbol{\sigma} \cdot \mathrm{d}\boldsymbol{a}], \tag{7.6}$$

where $[\boldsymbol{\sigma}\cdot\boldsymbol{a}] = [\boldsymbol{\sigma}\cdot\boldsymbol{n}]$ is the pressure drop across the interface Γ. The first two integrals can be rewritten using Gauss' theorem yielding

$$\int_{\partial\Omega_1} \boldsymbol{\sigma}\cdot \mathrm{d}\boldsymbol{a} = \int_{\Omega_1} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{r} \quad \text{and} \quad \int_{\partial\Omega_2} \boldsymbol{\sigma}\cdot \mathrm{d}\boldsymbol{a} = \int_{\Omega_2} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{r}, \tag{7.7}$$

respectively. The last integral can be rewritten using the Young-Laplace theorem

$$\int_{\Gamma} [\boldsymbol{\sigma}\cdot \mathrm{d}\boldsymbol{a}] = \int_{\Gamma} \gamma\kappa\,\mathrm{d}\boldsymbol{a}, \tag{7.8}$$

where γ is the surface tension and κ is the average curvature of the interface [5].

It is desirable to rewrite the surface integral (7.8) as a volume integral like the rest of the terms. This can be achieved by introducing a level set function $\phi(\boldsymbol{r},t)$ as we will show in the following. The proof will only be given in two dimensions but can easily be generalized to higher dimensions.

We introduce a level set function $\phi(\boldsymbol{r},t)$ with the properties

$$\begin{cases} \phi(\boldsymbol{r},t) > 0 & \boldsymbol{r}\in\Omega_1, \\ \phi(\boldsymbol{r},t) = 0 & \boldsymbol{r}\in\Gamma, \\ \phi(\boldsymbol{r},t) < 0 & \boldsymbol{r}\in\Omega_2. \end{cases} \tag{7.9}$$

This function uniquely defines the interface as $\Gamma = \{\boldsymbol{r}|\phi(\boldsymbol{r},t) = 0\}$ and lets us distinguish each subdomain by determining the sign of ϕ. We also introduce a transverse level set function $\psi(\boldsymbol{r},t)$ such that

$$\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}\psi = 0, \quad |\boldsymbol{\nabla}\psi| \neq 0. \tag{7.10}$$

We will show later that it is possible to construct such level set functions. Even though the method is generally applicable to systems of higher dimensions we will in the following consider a two dimensional system with the spacial dimensions x and y. We can construct a global orientation-preserving diffeomorphism that maps $\Omega \mapsto \Omega'$ through the variable change

$$x' = \psi(x,y) \tag{7.11}$$

$$y' = \phi(x,y). \tag{7.12}$$

This change of variables is area preserving because the Jacobian is non-zero,

$$\left|\frac{\partial(\psi,\phi)}{\partial(x,y)}\right| = (\phi_y, -\phi_x)\cdot(\psi_x,\psi_y) = |\boldsymbol{\nabla}\phi||\boldsymbol{\nabla}\psi| \neq 0, \tag{7.13}$$

where we assume that ψ is constructed such that $\boldsymbol{\nabla}\psi$ is parallel to the tangent direction and therefore $-\hat{\boldsymbol{\nabla}}\phi||\boldsymbol{\nabla}\psi$.

Furthermore we introduce a parameterization $\big(\overline{x}(s),\overline{y}(s)\big)$ of Γ, where s is an arc-length variable. Using this parameterization an infinitesimal change in x' along Γ is given by

$$\begin{aligned} \mathrm{d}x'|_{\phi=0} &= \mathrm{d}\psi(\overline{x}(s),\overline{y}(s)) \\ &= (\psi_x\overline{x}_s + (\psi_y\overline{y}_s)\,\mathrm{d}s \\ &= \big((\psi_x,\psi_y)\cdot(\overline{x}_s,\overline{y}_s)\big)\,\mathrm{d}s \\ &= |\boldsymbol{\nabla}\psi|\,\mathrm{d}s, \end{aligned} \tag{7.14}$$

where we have utilized the above assumption that the gradient of ψ is parallel to the tangent direction. With the above definitions we can rewrite Eq. (7.8) as

$$\begin{aligned}\int_\Gamma \gamma\kappa\,\mathrm{d}\boldsymbol{a} &= \int_{\phi=0} \gamma\kappa\boldsymbol{n}\,\mathrm{d}s \\ &= \int_{\phi=0} \gamma\kappa\frac{1}{|\boldsymbol{\nabla}\psi|}\boldsymbol{n}\,\mathrm{d}x' \\ &= \int_{\phi=0} \gamma\kappa\frac{1}{|\boldsymbol{\nabla}\psi|}\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\,\mathrm{d}x' \\ &= \int_{\Omega'} \gamma\kappa\delta(y')\frac{1}{|\boldsymbol{\nabla}\psi|}\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\,\mathrm{d}x'\,\mathrm{d}y',\end{aligned} \tag{7.15}$$

where we have used that the normal $\boldsymbol{n}$ to the interface can be written as $\boldsymbol{\nabla}/|\boldsymbol{\nabla}|$. Using Eq. (7.13) for changing variables, Eq. (7.15) becomes

$$\int_\Gamma \gamma\kappa\,\mathrm{d}\boldsymbol{a} = \int_\Omega \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi\,\mathrm{d}x\,\mathrm{d}y. \tag{7.16}$$

Inserting Eqs. (7.7) and (7.16) into Eq. (7.6) yields

$$\begin{aligned}\int_\Omega \rho\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t}\,\mathrm{d}\boldsymbol{r} &= \int_{\Omega_1} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{r} + \int_{\Omega_2} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{r} + \int_\Omega \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi\,\mathrm{d}\boldsymbol{r} \\ &= \int_\Omega \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi\,\mathrm{d}\boldsymbol{r}.\end{aligned} \tag{7.17}$$

This must hold true for any volume Ω. Hence

$$\rho\big[\partial_t\boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\big] = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi. \tag{7.18}$$

This is the level set formulation of the Navier-Stokes equation. We now need to show that we can construct the transverse level set function ψ with the required properties.

We start by defining a coordinate transformation by

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\Big(x(s,\tau), y(s,\tau)\Big) = \boldsymbol{\nabla}\phi\Big(x(s,\tau), y(s,\tau)\Big), \tag{7.19}$$

where

$$\Big(x(s,0), y(s,0)\Big) = \Big(\overline{x}(s), \overline{y}(s)\Big). \tag{7.20}$$

Because of the δ function in Eq. (7.18), ψ only needs to fulfill the requirements in a small region $|\tau| < \epsilon$ around Γ. In this small region we can define ψ as

$$\psi\Big(x(s,\tau), y(s,\tau)\Big) = \psi_0(s), \tag{7.21}$$

where $\psi_0(s)$ is a smooth increasing function if and only if the mapping of (x,y) to (s,τ) is one-to-one. Using the change of variables theorem we have to show that

$$\left|\frac{\partial(x,y)}{\partial(s,\tau)}\right| \neq 0. \tag{7.22}$$

Taylor expanding Eq. (7.19) around $\tau = 0$ yields

$$(x_\tau, y_\tau) = \boldsymbol{\nabla}\phi\Big(\overline{x}(s), \overline{y}(s)\Big) + \mathcal{O}(\tau). \tag{7.23}$$

Differentiation of Eq. (7.19) with respect to s and integration with respect to τ yields

$$\int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\frac{\mathrm{d}}{\mathrm{d}\tau'}\Big(x(s,\tau'), y(s,\tau')\Big)\,\mathrm{d}\tau' = \int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\nabla}\phi\Big(x(s,\tau'), y(s,\tau')\Big)\,\mathrm{d}\tau'. \tag{7.24}$$

Consequently,

$$\Big(x_s(s,\tau), y(s,\tau)\Big) - \Big(x_s(s,0), y_s(s,0)\Big) = \int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\nabla}\phi\Big(x(s,\tau'), y(s,\tau')\Big)\,\mathrm{d}\tau', \tag{7.25}$$

and thus

$$\begin{aligned}\Big(x_s(s,\tau), y(s,\tau)\Big) &= \Big(\overline{x}_s(s), \overline{y}_s(s)\Big) + \int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\nabla}\phi\Big(x(s,\tau'), y(s,\tau')\Big)\,\mathrm{d}\tau' \\ &= \mathbf{T}(s) + \mathcal{O}(\tau).\end{aligned} \tag{7.26}$$

Here $\mathbf{T}$ is a unit tangent vector to the interface. We can now calculate the determinant (7.22)

$$\begin{aligned}\left|\frac{\partial(x,y)}{\partial(s,\tau)}\right| &= (x_\tau, y_\tau)\cdot(-y_s, x_\tau) \\ &= \boldsymbol{\nabla}\phi(\overline{x}_s, \overline{y}_s)\cdot\hat{\mathbf{T}} \\ &= |\boldsymbol{\nabla}\phi||\mathbf{T}| + \mathcal{O}(\tau) \\ &= |\boldsymbol{\nabla}\phi|_{\phi=0} + \mathcal{O}(\tau) \neq 0.\end{aligned} \tag{7.27}$$

This means that ψ is well defined in a small region around Γ. Now all we need to prove is that $\boldsymbol{\nabla}\phi$ and $\boldsymbol{\nabla}\psi$ are orthogonal and that $|\boldsymbol{\nabla}\psi| \neq 0$. The orthogonality can be proved by differentiating ψ with respect to τ

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\psi\Big(x(s,\tau), y(s,\tau)\Big) &= \psi_x x_\tau + \psi_y y_\tau \\ &= \boldsymbol{\nabla}\psi\cdot\boldsymbol{\nabla}\phi = \frac{\mathrm{d}\psi_0(s)}{\mathrm{d}\tau} = 0,\end{aligned} \tag{7.28}$$

which means that ϕ and ψ are orthogonal if and only if $|\boldsymbol{\nabla}\psi| \neq 0$. This follows immediately from differentiating ψ with respect to s

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}s}\psi\Big(x(s,\tau), y(s,\tau)\Big) &= \psi_x x_s + \psi_y y_s \\ &= \boldsymbol{\nabla}\psi\cdot(x_s, y_s) \\ &= \boldsymbol{\nabla}\psi\cdot\mathbf{T} \\ &= |\boldsymbol{\nabla}\psi| = \psi_0'(s) > 0,\end{aligned} \tag{7.29}$$

because $\psi_0(s)$ was chosen to be an increasing function. Thereby we have established the level set formulation of the Navier-Stokes equation for a two liquid flow of incompressible fluids.

Because we are working in the microregime where the velocity of the flow is much smaller than the velocity of the sound (propagation of pressure), we can regard the fluid as being incompressible. This is, as described in chapter 2, formulated in the incompressibility condition

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0. \tag{7.30}$$

Now all we need in order to have the system completely described by dynamical equations is an equation describing the evolution of the zero level set. We only need to consider the movement of the zero level set because this is the only part of the level set function with a physical interpretation. Evolving the equation $\phi(\boldsymbol{r}, t) = 0$ in time defines the movement of the front. Differentiating with respect to time yields

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\phi(\boldsymbol{r}, t) = 0 \iff \\ \partial_t \phi(\boldsymbol{r}, t) + \boldsymbol{V} \cdot \boldsymbol{\nabla}\phi(\boldsymbol{r}, t) = 0, \end{aligned} \tag{7.31}$$

where $\boldsymbol{V} = \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t}\Big|_{\boldsymbol{r}\in\Gamma}$ is the velocity of the zero level set.

The velocity field must be continious at the particle surface. This is obtained by setting the convection velocity equal to the velocity of the fluid, $\boldsymbol{V} = \boldsymbol{u}$. The evolution equation for ϕ therefore becomes

$$\phi_t + \boldsymbol{u} \cdot \boldsymbol{\nabla}\phi = 0. \tag{7.32}$$

7.3 Smearing out the interface

In a system consisting of two immiscible viscid fluids (or a particle in a fluid) the density and viscosity are constant on each side of the interface. We can therefore define the dimensionless density and viscosity as

$$\tilde{\rho} = 1 + H(\phi)\left(\frac{\rho_1}{\rho_2} - 1\right) \tag{7.33}$$

and

$$\tilde{\eta} = 1 + H(\phi)\left(\frac{\eta_1}{\eta_2} - 1\right), \tag{7.34}$$

where $H(\phi)$ is a Heaviside function defined as

$$H(\phi) = \begin{cases} 1, & \phi \in \Omega_1 \\ 0, & \phi \in \Omega_2. \end{cases} \tag{7.35}$$

Setting $\rho_0 = \rho_2$ ensures that the density of the fluid is ρ_1 and ρ_2 in Ω_1 and Ω_2, respectively. Similarly setting $\eta_0 = \eta_2$ makes the viscosity of the fluid η_1 and η_2 in Ω_1 and Ω_2, respectively.

When solving the system numerically the abrupt change in density and viscosity across the interface causes numerical instabilities to occur. In order to avoid this we need to smear out the Heaviside function, delta function and sign function across the interface. We do this by substituting $H(\phi)$, $\delta(\phi)$ and $\text{sign}(\phi)$ with the smeared out versions $H_\epsilon(\phi)$, $\delta_\epsilon(\phi)$ and $\text{sign}_\epsilon(\phi)$ defined as

$$H_\epsilon(\phi) \equiv \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{\phi}{\epsilon}\right), \tag{7.36a}$$

$$\delta_\epsilon(\phi) \equiv H'_\epsilon(\phi) = \frac{1}{2\epsilon} - \frac{1}{2\epsilon}\tanh^2\left(\frac{\phi}{\epsilon}\right) \quad \text{and} \tag{7.36b}$$

$$\text{sign}_\epsilon(\phi) \equiv \tanh\left(\frac{\phi}{\epsilon}\right) \tag{7.36c}$$

respectively. This implies that the interface has a finite thickness Γ_ϵ of approximately

$$\Gamma_\epsilon = \frac{2\epsilon}{|\boldsymbol{\nabla}\phi|}. \tag{7.37}$$

It is necessary to maintain a uniform thickness of the interface throughout the calculations. This requires that the gradient of the level set function is constant within a region around the interface $|\phi| < \epsilon$. This is not automatically fulfilled. The time evolution of any level set $\phi(\boldsymbol{r}, t) = C$ is given by the level set equation (7.32). This means that the height of the level set function will remain constant, but it does not ensure that the gradient does not change. Thus in order to keep a fixed interface thickness we need to reinitialize the level set function without changing the zero level set. As long as the reinitialization procedure does not change the zero level set the physics of the problem is unaffected.

7.4 Signed distance functions

Until now we have not considered which function to use as the level set function. The only requirement is that it is smooth and that it uniquely identifies the front which we want to evolve as its zero level set. Moreover each side of the front should be identified by the sign of the level set function. In principle we can use any function that fulfills (7.9) since only the zero level set has a physical interpretation. But requiring the interface thickness to be fixed constrains the gradient of ϕ to be fixed in a region around the interface. A choice of $\phi(\boldsymbol{r}, t)$ that fulfills this requirement is the signed distance function, where the distance is the shortest distance $d(\boldsymbol{r})$ from a point to the interface

$$d(\boldsymbol{r}) = \pm\min(|\boldsymbol{r} - \boldsymbol{r}_\Gamma|), \tag{7.38}$$

where $\boldsymbol{r}_\Gamma$ is the points on the interface. The minus sign applies if $\boldsymbol{r} \in \Omega_1$ and the plus sign if $\boldsymbol{r} \in \Omega_2$.

This choice of level set function has a number of special properties [20, 22]. Amongst those, the length of the gradient

$$|\boldsymbol{\nabla}\phi| = 1, \tag{7.39}$$

the unit normal vector to the interface

$$\boldsymbol{n} = \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}, \tag{7.40}$$

and the curvature of the zero level set

$$\kappa = \boldsymbol{\nabla} \cdot \boldsymbol{n} = \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|} \right). \tag{7.41}$$

These properties must be invariant under the reinitialization procedure.

7.5 Reinitialization

Several approaches have been suggested in order to maintain the level set function as a signed distance function. Common for all methods is that the time evolution of the physical problem is stopped and the level set function is straightened out. The position of the zero level set must be conserved during the reinitialization procedure in order to conserve the physics of the problem. When the level set function has been reinitialized, the time-evolution of the physical problem is resumed using the reinitialized level set function.

Simple approach

The simplest way to reinitialize the level set function as a signed distance function is simply to stop the computation when the level set function is too distorted and compute the shortest distance from every point in the computational domain to the interface.

This method has been used by Chopp [7], but it is not very attractive because it is computationally very expensive.

Reinitialization equation

Another more sophisticated approach was suggested by Sussman, Smereka and Osher [23]. They reinitialized the level set function using the the partial differential equation

$$\partial_\tau \psi(\mathbf{r}, \tau) = \mathrm{sign}(\phi)(1 - |\boldsymbol{\nabla}\psi(\mathbf{r}, \tau)|). \tag{7.42}$$

The equation is solved to steady state with the initial condition $\psi(\mathbf{r}, 0) = \phi$. τ is a pseudo-time used only during the reinitialization. The sign function changes sign across the interface and is zero at the interface. Thus, the interface are fixed at the same position because $\mathrm{sign}(\phi) = 0$ at the interface making $\partial_\tau \psi = 0$ at the interface. Away from the interface ψ will change until the length of the gradient equals 1, which as stated above is a signed distance function.

Because numerical oscillations can occur if the sign of ϕ changes abruptly at the interface, it is necessary to use the smeared out sign function given in Eq. (7.36c).

Using this method mass loss can still occur due to numerical diffusion. Sussmann and Fatemi [23] therefore proposed a correction to the above reinitialization equation

$$\partial_\tau \psi(\mathbf{r}, \tau) = \mathrm{sign}(\phi)(1 - |\boldsymbol{\nabla}\psi(\mathbf{r}, \tau)|) + \lambda f(\psi), \tag{7.43}$$

where $f(\psi) = \delta(\psi)|\boldsymbol{\nabla}\psi|$.

We want the volume of the particle to be constant. Hence we demand that

$$\partial_t \int_\Omega H(\psi)\, \mathrm{d}\mathbf{a} = 0. \tag{7.44}$$

Moving the differentiation under the integral sign yields

$$\int_\Omega \delta(\psi)\psi_t \,\mathrm{d}\mathbf{a} = 0 \tag{7.45}$$

Inserting the modified partial differential equation for ϕ gives us the requirement that must be fulfilled

$$\int_\Omega \delta(\psi)\big(\operatorname{sign}(\phi)(1 - |\boldsymbol{\nabla}\psi(\mathbf{r},\tau)|) + \lambda f(\psi)\big)\,\mathrm{d}\mathbf{a} = 0. \tag{7.46}$$

We have chosen to use this reinitialization procedure for the simulations in this thesis. In section 3.2 we describe how we have implemented the system of governing equations and the reinitialization procedure in Femlab. The volume conserving constraint is implemented using a weak constraint via the field `fem.equ.constr`.

Tracer particles

A third way of reinitializing the level set function is described by Enright *et al.* [10]. Enright *et al.* passively advect tracer particles with the fluid. The tracer particles must always stay on the same side of the zero level set. If a marker particle crosses the front, it means that the level set function has been distorted and must be rebuild. The level set function is then reconstructed such that the marker particles are again on the correct side of the front.

Promising results has been obtained using this method, but it is not straight forward to implement in Femlab. A possible route for implementation could build on the tools developed by Wu Zhilei in his doctoral thesis “Numerical Study of Dispersed Two-phase Flows” (Lund Institute of Technology, Sweden, 2000)[1].

Extension velocities

Equation (7.31) is the general equation describing the time evolution of the level set function. In fluidic problems, we use the velocity field of the fluid as the convection velocity field in order to fulfill the no-slip condition on the surface of the particle. However, only the velocity of the zero level set $\boldsymbol{V}(\phi = 0)$ is relevant for the physical problem. In some physical problems this can be exploited so that reinitialization becomes unnecessary. If the velocity of the front is orthogonal to the front itself then $\boldsymbol{V}(\phi = 0)\|\boldsymbol{\nabla}\phi$ and Eq. (7.31) reduces to

$$\phi_t + V|\boldsymbol{\nabla}\phi| = 0 \tag{7.47}$$

in the region near the evolving front. Instead of setting $\boldsymbol{V}$ equal to the physical velocity field $\boldsymbol{u}$, Adalsteinsson and Sethian [1] define a so called extension velocity $\boldsymbol{V}_\mathrm{ext}$ that fulfills the requirement of the physical problem $\boldsymbol{V}_\mathrm{ext}(\phi = 0) = \boldsymbol{u}$ and the equation

$$\boldsymbol{\nabla}V_\mathrm{ext} \cdot \boldsymbol{\nabla}\phi = 0. \tag{7.48}$$

If the level set function is initialized as a signed distance function with $|\boldsymbol{\nabla}\phi(\boldsymbol{r}, 0)| = 1$ then it will remain so at all times because

$$\begin{aligned}\frac{\mathrm{d}|\boldsymbol{\nabla}\phi|^2}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{\nabla}\phi \cdot \boldsymbol{\nabla}\phi\right) = 2\boldsymbol{\nabla}\phi \cdot \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\nabla}\phi \\ &= -2\boldsymbol{\nabla}\phi \cdot \boldsymbol{\nabla}V_\mathrm{ext}|\boldsymbol{\nabla}\phi| - 2\boldsymbol{\nabla}\phi \cdot \boldsymbol{\nabla}|\boldsymbol{\nabla}\phi|V_\mathrm{ext}.\end{aligned} \tag{7.49}$$

[1] `http://www.comsol.dk/showroom/gallery/314.php`

The first term on the right hand side of the above equation is zero because of Eq. (7.48), and the second term is zero because $|\boldsymbol{\nabla}\phi(\boldsymbol{r},0)| = 1$ which reduces the second term on the right hand side to $-2\boldsymbol{\nabla}\phi \cdot \boldsymbol{\nabla}V_{\text{ext}} = 0$, and the length of $\boldsymbol{\nabla}\phi$ is therefore 1 at all times.

Though this method elegantly preserves the level set function as a signed distance function, it is not implementable in FEMLAB because we need to solve Eq. (7.48) under the constraint that the velocity on the zero level set equals the velocity of the fluid. This is not possible because FEMLAB offers no way of formulating a constraint implicitly.

Furthermore, the method is not applicable in the present problem because the velocity of the front is not orthogonal to the front itself.

7.6 Implementation

One of the great advantages of the level set formulation is that we do not track an interface explicitly but rather capture it implicitly. This makes several numerical tools applicable for solving the dynamical system. In this chapter we describe how we have implemented the method in FEMLAB.

7.7 Nondimensionalizing the equations

Again we introduce characteristic scales for the physical quantities in the problem. We define the characteristic length scale L_0, velocity scale U_0, density ρ_0, viscosity η_0 and surface tension γ_0. Expressing the physical quantities as a dimensionless number times the characteristic scale we find

$$\mathbf{r} = L_0\tilde{\mathbf{r}}, \quad \boldsymbol{u} = U_0\tilde{\boldsymbol{u}}, \quad \rho = \rho_0\tilde{\rho}, \quad \eta = \eta_0\tilde{\eta} \quad \text{and} \quad \gamma = \gamma_0\tilde{\gamma}, \tag{7.50}$$

where the dimensionless quantities are denoted by a tilde. Similarly we can define the characteristic pressure and timescale as relations between the chosen characteristic parameters

$$p = \frac{\eta_0 U_0}{L_0}\tilde{p} \quad \text{and} \quad t = \frac{L_0}{U_0}\tilde{t}. \tag{7.51}$$

Substituting (7.50) and (7.51) into Navier-Stokes equation (7.18) yields

$$Re\tilde{\rho}\left(\frac{\partial\tilde{\boldsymbol{u}}}{\partial\tilde{t}} + (\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{\nabla}})\tilde{\boldsymbol{u}}\right) = (\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{\sigma}}) + \frac{1}{Ca}\tilde{\gamma}\tilde{\kappa}\delta(\phi)\tilde{\boldsymbol{\nabla}}\phi \tag{7.52}$$

where the Reynolds number $Re = \rho_0 U_0 L_0/\eta_0$ is the ratio between inertial forces and viscous forces and the Capillary number $Ca = \eta_0 U_0/\gamma_0$ is the ratio between viscous forces and the surface tension forces.

The dimensionless form of the continuity equation and the level set equation is

$$\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{u}} = 0 \tag{7.53}$$

and

$$\phi_t + \tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{\nabla}}\phi = 0 \tag{7.54}$$

respectively. We note that because of the continuity condition the convection equation for the level set function can be rewritten as

$$\phi_t + \tilde{\boldsymbol{\nabla}}(\phi\tilde{\boldsymbol{u}}) = 0 \tag{7.55}$$

which is a more useful form when implementing the equation into Femlab.

7.8 Implementation in Femlab

In the following we will go through the steps taken to implement the level set formulation of a microfluidic system with a finite-sized particle passively convected by the flow. The solution process consists of a series of steps as illustrated in figure 7.2

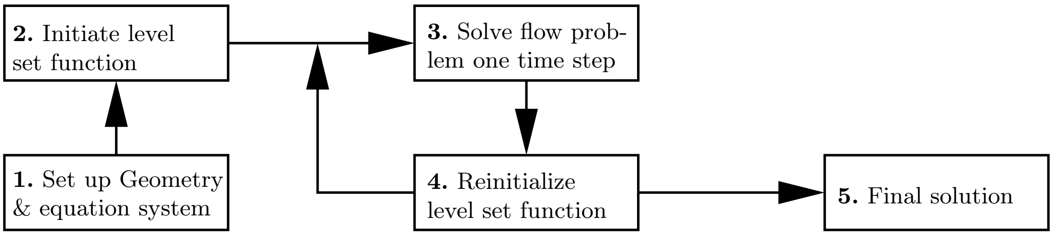

2. Initiate level set function
3. Solve flow problem one time step
1. Set up Geometry & equation system
4. Reinitialize level set function
5. Final solution

Figure 7.2: Block diagram of the steps in the level set solution process. 1. The channel geometry and the equation system is initiated. 2. The level set function is initiated as a signed distance function. 3. The flow problem and the convection of the level set function is solved for one time step. The reinitialized level set function (or the initially defined signed distance function) is used as initial condition for the level set function. 4. The level set function is reinitialized. Step no. 3. and 4. is repetend until 5. The final solution is obtained.

Setting up the equations

We use the general equation form described in chapter 3 to set up the equations. In the level set formulation, we solve the system as a time-dependent system directly in Femlab. The quantity d_{a_i} is therefore nonzero. To put equations (7.52), (7.54) and (7.55) into Femlab we use

$$d_{a_1} = Re\tilde{\rho}, \qquad d_{a_2} = Re\tilde{\rho}, \qquad d_{a_3} = 0, \quad d_{a_4} = 1, \tag{7.56a}$$

$$\boldsymbol{\Gamma}_1 = \begin{bmatrix} -\tilde{\sigma}_{11} \\ -\tilde{\sigma}_{21} \end{bmatrix}, \quad \boldsymbol{\Gamma}_2 = \begin{bmatrix} -\tilde{\sigma}_{12} \\ -\tilde{\sigma}_{22} \end{bmatrix}, \quad \boldsymbol{\Gamma}_3 = \mathbf{0}, \quad \boldsymbol{\Gamma}_4 = \mathbf{0}, \tag{7.56b}$$

$$\begin{aligned} F_1 &= -Re\tilde{\rho}(\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{\nabla}})\tilde{u}_1 + \frac{1}{Ca}\tilde{\gamma}\tilde{\kappa}\delta(\phi)\tilde{\partial}_x\phi, \quad F_3 = \tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{u}}, \\ F_2 &= -Re\tilde{\rho}(\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{\nabla}})\tilde{u}_2 + \frac{1}{Ca}\tilde{\gamma}\tilde{\kappa}\delta(\phi)\tilde{\partial}_y\phi, \quad F_4 = -\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{\nabla}}\phi, \end{aligned} \tag{7.56c}$$

where the solution vector is $\boldsymbol{U} = [\tilde{u}_1, \tilde{u}_2, \tilde{p}, \phi]$. The dimensionless velocity components are denoted by $\tilde{u}_1$ and $\tilde{u}_2$, respectively, and the dimensionless pressure is denoted by $\tilde{p}$. The level set function ϕ is initialized as the dimensionless signed distance to the particle-liquid interface.

The stress tensor components $\tilde{\sigma}_{ij}$ and the curvature of the surface $\tilde{\kappa}$ are defined as subdomain expressions

$$\tilde{\sigma}_{11} = 2\tilde{\eta}\tilde{\partial}_x\tilde{u}_1 - \tilde{p}, \qquad \tilde{\sigma}_{12} = \tilde{\eta}(\tilde{\partial}_x\tilde{u}_2 + \tilde{\partial}_y\tilde{u}_1), \tag{7.57}$$

$$\tilde{\sigma}_{21} = \tilde{\sigma}_{12}, \qquad \tilde{\sigma}_{22} = 2\tilde{\eta}\tilde{\partial}_y\tilde{u}_2 - \tilde{p} \tag{7.58}$$

and

$$\tilde{\kappa} = \frac{\tilde{\partial}_{xx}\phi(\tilde{\partial}_y\phi)^2 - 2\tilde{\partial}_x\phi\tilde{\partial}_y\phi\tilde{\partial}_{xy}\phi + \tilde{\partial}_{yy}\phi(\tilde{\partial}_x\phi)^2}{\big((\tilde{\partial}_x\phi)^2 + (\tilde{\partial}_y\phi)^2\big)^{\frac{3}{2}}} \tag{7.59}$$

The above defines the dynamical problem. The reinitialization procedure is implemented in a similar fashion

$$d_a = 1, \tag{7.60a}$$

$$\mathbf{\Gamma} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{7.60b}$$

$$F = \operatorname{sign}_\epsilon(\phi_0)\left(1 - \sqrt{(\tilde{\partial}_x\psi)^2 + (\tilde{\partial}_y\psi)^2}\right), \tag{7.60c}$$

where ϕ_0 is the level set function from the last time step of the dynamical problem. The volume constraint is implemented as

$$0 = V_0 - V(\psi), \tag{7.61}$$

where $V_0 = \int_\Omega \mathrm{d}\boldsymbol{r}\left(1 - H_\epsilon\big(\phi(t=0)\big)\right)$ is the initial volume of the particle and $V(\psi) = \int_\Omega \mathrm{d}\boldsymbol{r}\left(1 - H_\epsilon\big(\psi(\tau)\big)\right)$ is the current volume of the particle. The integrals are computed using the integrino coupling variables provided by FEMLAB.

7.9 An example

In the following section we will use the derived level set formulation. As a case study we examine the path of a two dimensional particle when it is convected by a fluid past an obstacle.

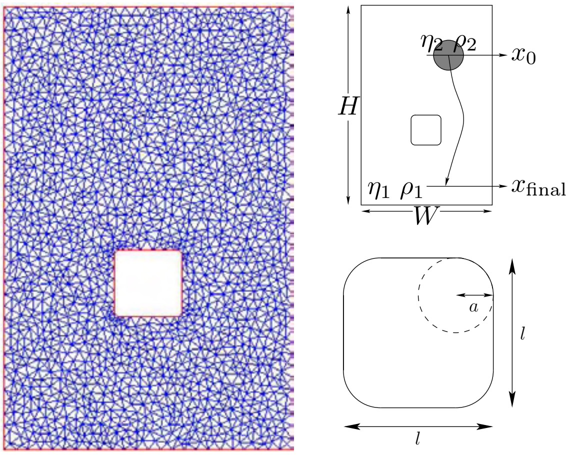


Figure 7.3: For the test study we use the geometry consisting of a section of a channel with a single obstacle. The shape of the obstacle was changed from simulation to simulation by changing the radius of curvature a for the corner of the obstacle. A series of simulations were run changing the shape of the obstacle from round ($a = l/2$) to square ($a = 0$). The height of the channel is $H = (20/3)l$ and the width of the channel section is $W = (13/3)l$. The starting position x_0 and the final position x_{final} of the particle were recorded. For each geometry configuration a mesh similar to the one shown on the right was generated and kept fixed throughout the simulation.

Geometry and mesh

The geometry used for this test study is simply a channel with an obstacle as shown in figure 7.3. We ran a series of simulations with the shape of the obstacle changing from circular to quadratic. For each simulation we constructed a geometry and generated a mesh. The mesh and geometry were fixed throughout the entire simulation.

The parameters used in the simulation are as tabulated in Tab. 7.1.

The flow in the channel is driven by a pressure drop Δp from top to bottom of the channel. The boundary conditions used in the simulations are no-slip at the solid obstacle, no-stress conditions at the sides of the computational domain and at the bottom wall the pressure is fixed to zero. The boundary conditions are implemented as discussed in section 3.2.

The particle is represented by the negative part of the level set function and the fluid resides in the part of the computational domain where the level set function is positive. We study nonbuoyant particles, so the density of the particle is the same as the density of the submersing fluid. The viscosity of the particle is 100 times higher than the viscosity of the fluid. The particle is initiated as a circular particle vertically situated at $y_0 = H - l$. For each geometry, we used four different initial horizontal positions x_0. When the center of a convected particle passes the horizontal line positioned at $y = l$ the final horizontal position x_{final} is detected. In this formulation the level set function is initiated as the signed distance function given by

$$\phi(x, y, t = 0) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - R_{\text{P}}. \tag{7.62}$$

The algorithm for solving the equation system described above is

- Initiate the level set function as a signed distance function based on the shape of the particle.
- Evolve the level set function one time step according to the level set equation (7.55) using the velocity field and pressure computed from Navier-Stokes equation (7.52) and the continuity equation (7.54).
- Reinitialize the level set function without changing the position of the zero level set. We use the reinitialization equation (7.46) where we constrain the volume to be constant using a weak constraint.

Table 7.1: The parameter values used in the test case.

Reynolds number	Re	=	1×10^{-3}
Capillary number	Ca	=	1×10^{6}
Density	ρ_0	=	$1 \times 10^{3}\,\text{kg·m}^{-3}$
Viscosity	η_0	=	$1 \times 10^{-1}\,\text{Pa·s}$
Obstacle size	l	=	$6 \times 10^{-6}\,\text{m}$
Particle radius	R_{P}	=	$3 \times 10^{-6}\,\text{m}$
Pressure drop	Δp	=	$1.2 \times 10^{-3}\,\text{Pa}$
Time step	Δt	=	$5 \times 10^{-2}\,\text{s}$
Mesh element size	`hmesh`	=	$1.1 \times 10^{-6}\,\text{m}$
Thickness parameter	ϵ	=	$0.5 \times$ `hmesh`

- Use the reinitialized level set function and evolve the system one more time-step.

We have included code snippets in Appendix C to show how each step is implemented.

Some results

We carried out simulations for four different initial positions of the particle. The normalized initial horizontal positions $2x_0/W$ were 1.015, 1.077, 1.308 and 1.539, respectively. The center of the channel is $x = W/2$. For each of these initial positions, we ran a series of simulations with five different shapes of the obstacle. The radii for the rounded corner of the obstacle were $2a/l = i/10$, with $i = 1, 3, 5, 7, 10$.

For each combination of initial position and obstacle shape we solved the system and obtained the particle paths. Snapshots of the particles positions are shown in Figs. 7.5 and 7.6 for the round and square obstacle, respectively.

The difference in the horizontal position Δx as a function of obstacle shape is plotted in figure 7.4. As is evident, the results are not as systematically accurate as we would have liked. However some of the expected behaviors can be observed:

- Particles started near the center line of the channel have a larger displacement than particles started further away from the center line.
- The round obstacle tend not to drag as much in the particles as square obstacles yielding a larger displacement from start to finish.
- Particles started near the center line of the channel have a greater probability of getting stuck in the dead zone in front of the obstacle. Only one of the particles

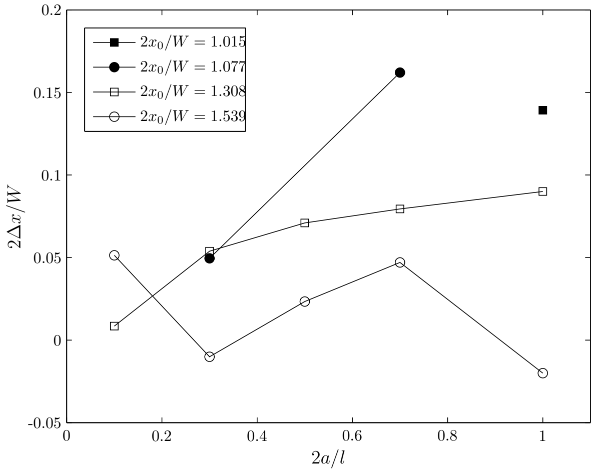


FIGURE 7.4: For particles passing obstacles of different shapes normalized difference $2\Delta x/W$ in horizontal position from start to finish is plotted versus starting position $2a/l$. The missing data points for the simulations with the initial positions of the particles nearest to the center of the channel is due to the particles getting being stuck at the obstacle and hence not reaching the final position.

> started closest to the center line of the channel got past the obstacle. This was the one passing the round obstacle.

Thus, it was possible to implement the level set method in Femlab, and the obtained results showed qualitative agreement with the expected outcome. However, the results did not provide the accuracy needed to do simulations of particle transport in microsystems for precise determination of geometrical properties of the particles.

This is partly due to the reinitialization procedure. The constraint used when reinitializing the level set function is a constraint over the entire computational domain. This allows the zero level set to move. A better approach would be a local constraint on the mesh cells intercepted by the front. Unfortunately, such local constraints are not possible in Femlab. Further more, the constraints containing integrals causes the time stepper to take extremely small steps. Consequently, the solution times become very long and the solution process is very memory consuming. We were not able to run the simulations on an ordinary PC with 2 Gb RAM but had to use the GridEngine batch system available on the High Performance Computer (HPC) systems at DTU.

For investigation of the dependance of the morphology of finite-sized particles in microfluidic systems there is another inconvenience with this level set implementation. This is the preservation of shape. As long as we use spherical particles, the viscosity and surface tension will ensure that the particles do not lose shape. But this will not be the case for other shapes. They will gradually become spherical. To maintain the shape of a non-spherical particle, other reinitialization procedures must be used. One possibility is the particle level set method discussed earlier.

Nevertheless, the level set method in the implementation described in this thesis might prove useful for the simulation of cells and other biomolecules that are not completely rigid in microsystems. The level set method might also be valuable for simulating two phase flows – especially if the difference between the viscosity of the two fluids is not so large.

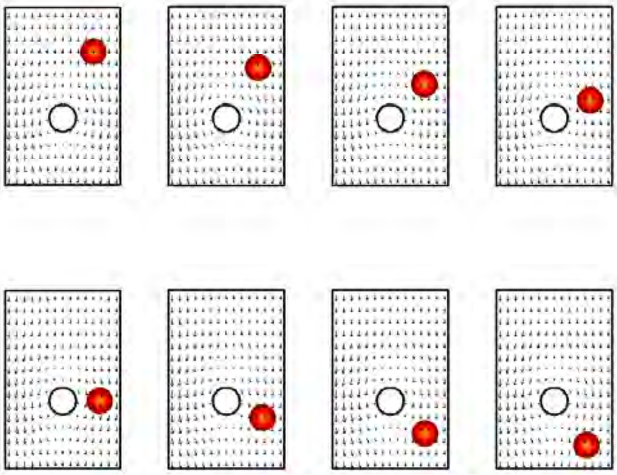


Figure 7.5: Results from the case study with round obstacle.

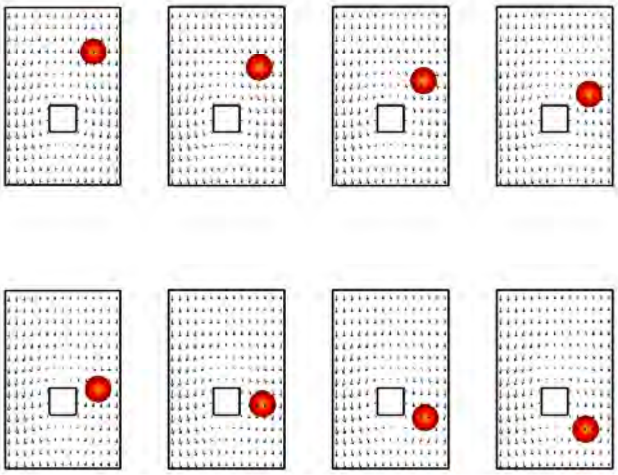


Figure 7.6: Results from the case study with almost square obstacle.

Chapter 8

Conclusion and outlook

In the present thesis we have presented what to our knowledge is the first numerical code for simulating the convection of finite-sized, arbitrarily shaped particles in microsystems. Our implementation enables simulation of arbitrarily shaped particles convected by a fluidic flow in narrow channel systems. The interaction forces between the particles and the solid walls of the containing channels are modeled as an added force per surface area of the particle if the particle is in the vicinity of the walls.

The model yields results that is in qualitative agreement with the expected outcome. The complexity and the size of the systems that can be modeled using out implementation is limited by the available computer power, but we have modeled quite complex geometries by utilizing the symmetries and periodic properties of the modeled devices. The implementation has been done in FEMLAB using the possibility to add features via the MATLAB scripting interface. This implementation ensures that systems with other governing physics can be easily modeled.

In an attempt to overcome some of the computer power demanding shortcomings of the presented implementation we investigated the level set method as a means of modeling particle transport without the need to rediscretize the computational domain every time step. The method proved to be a valuable method for simulating two phase flows. Solid particle transport can be viewed as a special case of two phase flows where the viscosity ratio between the two phases are large. With this model we were able to use the level set method to simulate the convection of a circular particle past a solid obstacle. We provided an implementation of a volume preserving constraint for circular particles, but our implementation were not able to conserve the shape of arbitrarily shaped solid particles. More work is needed in this area, but the current implementation might be useful for simulating soft particles, bubbles or biological cells in microsystems.

The future holds numerous promising possibilities for microsystems. The recent rapid development in the polymer sciences provides many exciting new areas of application and many new functionalities for microsystems. Especially in the field of lab-on-a-chip systems interesting new systems have been engineered in the last decade. Many of these systems are developed for medical or biological analysis. More often than not the samples of interest involves fluids and in many cases the investigation of submersed particles is the target. Here microfluidic chips are indispensable.

To develop new and more advanced systems in the future and innovate existing systems, an understanding of the dynamics of particles in microsystems is inevitable. The analysis and tools described in this thesis provides the basis for future studies of microsystems for handling of finite sizes particles.

The Femlab provided implementation makes the developed code applicable to systems where the particles are under influence of other forces than the contact forces from the channel walls and the hydrodynamic forces included in the models in this thesis. This could for example be externally applied magnetic or electrical forces but polymer based microchannels with active sites incorporated in the channelwalls are active areas of research and detailed numerical studies are required in order to design microchips that fully utilize the possibilities of such advanced systems.

Martin Heller
MIC – Department of Micro and Nanotechnology
Technical University of Denmark
July 31, 2005

Appendices

Appendix A

Submitted article

Title

Particle motion in microfluidics simulated using a Femlab implementation of the level set method.

Authors

Martin Heller and Henrik Bruus.

Paper

Submitted to Physical Review E May 4, 2005.

Particle motion in microfluidics simulated using a Femlab implementation of the level set method

Martin Heller and Henrik Bruus
MIC – Department of Micro and Nanotechnology, DTU bldg. 345 east
Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark

(Dated: June 28, 2005)

We implement the level set method for numerical simulation of the motion of a suspended particle convected by the fluid flow in a microchannel. The method automatically cope with the interactions between the particle and the channel walls. We apply the method in a study of particles moving in a channel with obstacles of different shapes. The generality of the method also makes it applicable for simulations of motion of particles under influence of external forces.

I. INTRODUCTION

In recent years numeral lab-on-a-chip systems have been developed to analyze biological samples. Many of these systems rely on handling of particles and cells comparable in size to the dimensions of the channels containing them. Examples of such microsystems are bumper-arrays or DEP-systems [1, 2, 3, 4]

It is a major challenge in theoretical microfluidics to study the dynamics of particles of finite size when they are convected by a fluid flow. Especially problematic is the forces appearing during collisions of the particles with the walls of the channel.

The level set method [5] is well suited to cope with these problems. By introducing a hypersurface $\phi(\mathbf{r},t)$, the particle interface is represented as the zero level set $\phi(\mathbf{r},t)=0$. The major advantage of the method is that this zero level set can be calculated implicitly instead of explicit tracking of the points on the interface.

The manuscript is organized as follows: In Sec. II we state the equations governing the dynamics of the system and in Sec. III we derive the level set formulation for the tracked interface. The implementation of the method in the numerical simulation tool Femlab is described in Sec. IV and we present results of a test study in Sec. VI. Finally, we evaluate the method in Sec. VII and give suggestions to future areas of usage.

II. GOVERNING EQUATIONS

We consider microfluidic systems. Hence the characteristic length scales of channels are of the order of 10 μm which is well beyond the intermolecular distances characteristic of the fluids involved. Thus the continuum hypothesis applies. Moreover, in these systems the flow velocities are much smaller than the propagation of pressure (the speed of sound). We can therefore consider the fluids to be incompressible and the continuity condition

$$\boldsymbol{\nabla}\cdot\mathbf{u}=0 \tag{1}$$

holds true for the velocity field $\mathbf{u}$ of the fluid.

Consider a domain Ω consisting of two subdomains Ω_1 and Ω_2 with surfaces $\partial\Omega_1$ and $\partial\Omega_2$, respectively. The common boundary between Ω_1 and Ω_2 is the interface Γ which we want to evolve.

The rate of change of the momentum of the fluid is given by $\int_\Omega \rho\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\,\mathrm{d}\mathbf{r}$ involving the substantial time derivative of $\mathbf{u}$. The change in momentum arises from the forces acting on the volume of fluid. In a microfluidic system we can neglect gravity and the only force $\mathbf{F}_\sigma$ acting on a volume of fluid Ω stems from the stresses $\boldsymbol{\sigma}$ exerted by the surrounding liquid on the surface $\partial\Omega$,

$$\mathbf{F}_\sigma=\int_{\partial\Omega}\boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{a}, \tag{2}$$

where $\boldsymbol{\sigma}$ is the stress tensor modelled by

$$\sigma_{ij}=-p\delta_{ij}+\eta\left(\partial_j u_i+\partial_i u_j\right). \tag{3}$$

Newton's second law therefore takes the form

$$\int_\Omega \rho\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\,\mathrm{d}\mathbf{r}=\int_{\partial\Omega}\boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{a}. \tag{4}$$

The right hand side of this equation can be split up in three integrals; two parts for each of the boundaries of the two subdomains and one along the common interface

$$\begin{aligned}\int_\Omega \rho\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\,\mathrm{d}\mathbf{r}&=\int_{\partial\Omega_1}\boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{a}+\int_{\partial\Omega_2}\boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{a}+\int_\Gamma[\boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{a}]\\&=\int_{\Omega_1}\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}\,\mathrm{d}\mathbf{r}+\int_{\Omega_2}\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}\,\mathrm{d}\mathbf{r}+\int_\Gamma\gamma\kappa\,\mathrm{d}\mathbf{a},\end{aligned} \tag{5}$$

In the second equality we have used Gauss' theorem as well as the Young–Laplace law relating the pressure drop $[\boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{a}]$ across the interface Γ to the surface tension γ and average curvature κ.

To facilitate numerical computation it is desirable to rewrite the last integral in Eq. (5) as a volume integral like the rest of the terms. This can be achieved by introducing a level set function $\phi(\mathbf{r},t)$ as we will show in the following.

III. THE LEVEL SET METHOD

Following Ref. [6] we introduce a level set function $\phi(\mathbf{r},t)$ with the properties

$$\begin{cases} \phi(\mathbf{r},t) > 0, & \mathbf{r} \in \Omega_1, \\ \phi(\mathbf{r},t) = 0, & \mathbf{r} \in \Gamma, \\ \phi(\mathbf{r},t) < 0, & \mathbf{r} \in \Omega_2. \end{cases} \tag{6}$$

This function uniquely defines the interface as $\Gamma(t) = \{\mathbf{r}|\phi(\mathbf{r},t) = 0\}$ and permits us to distinguish each subdomain by the sign of ϕ. We also introduce a transverse level set function $\psi(\mathbf{r},t)$ such that

$$\boldsymbol{\nabla}\phi \cdot \boldsymbol{\nabla}\psi = 0, \quad |\boldsymbol{\nabla}\psi| \neq 0. \tag{7}$$

We show in Appendix A that it is possible to construct such level set functions. In the following we consider a two dimensional system, but the method is applicable in higher dimensions also. We can construct a global orientation-preserving diffeomorphism that maps $\Omega \mapsto \Omega'$ through the variable change

$$x' = \psi(x,y) \tag{8a}$$

$$y' = \phi(x,y). \tag{8b}$$

We denote partial derivatives with indices, e.g., $\psi_x \equiv \partial_x\psi$. The change of variables Eqs. (8) is area preserving because the Jacobian is non-zero,

$$\left|\frac{\partial(\psi,\phi)}{\partial(x,y)}\right| = (\phi_y, -\phi_x)\cdot(\psi_x,\psi_y) = |\boldsymbol{\nabla}\phi||\boldsymbol{\nabla}\psi| \neq 0, \tag{9}$$

where we assume that ψ is constructed such that $\boldsymbol{\nabla}\psi$ is parallel to the tangent direction and therefore $-\boldsymbol{\nabla}\phi||\boldsymbol{\nabla}\psi$.

Furthermore we introduce a parameterization $(\overline{x}(s), \overline{y}(s))$ of Γ, where s is an arc-length variable. Using this parameterization an infinitesimal change in x' along Γ is given by

$$\mathrm{d}x'|_{\phi=0} = |\boldsymbol{\nabla}\psi|\,\mathrm{d}s, \tag{10}$$

where we have utilized the above assumption that the gradient of ψ is parallel to the tangent direction. With the above definitions we can rewrite the surface integral in Eq. (5) as

$$\begin{aligned} \int_\Gamma \gamma\kappa\,\mathrm{d}\mathbf{a} &= \int_{\phi=0} \gamma\kappa\mathbf{n}\,\mathrm{d}s \\ &= \int_{\phi=0} \gamma\kappa\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\frac{1}{|\boldsymbol{\nabla}\psi|}\,\mathrm{d}x' \\ &= \int_{\Omega'} \gamma\kappa\delta(y')\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\frac{1}{|\boldsymbol{\nabla}\psi|}\,\mathrm{d}x'\,\mathrm{d}y', \end{aligned} \tag{11}$$

where we have used that the normal $\mathbf{n}$ to the interface can be written as $\boldsymbol{\nabla}\phi/|\boldsymbol{\nabla}\phi|$. Using Eq. (9) for changing variables, Eq. (11) becomes

$$\int_\Gamma \gamma\kappa\,\mathrm{d}\mathbf{a} = \int_\Omega \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi\,\mathrm{d}x\,\mathrm{d}y. \tag{12}$$

Inserting Eq. (12) into Eq. (5) yields

$$\int_\Omega \rho\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\,\mathrm{d}\mathbf{r} = \int_\Omega [\boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi]\,\mathrm{d}\mathbf{r}. \tag{13}$$

This must hold true for any volume Ω. Hence

$$\rho\left[\partial_t\mathbf{u} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\right] = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \gamma\kappa\delta(\phi)\boldsymbol{\nabla}\phi, \tag{14}$$

which is the level set formulation of the Navier–Stokes equation.

In order to have the system completely described by dynamical equations we finally need an equation describing the evolution of the zero level set. We only need to consider the movement of the zero level set because this is the only part of the level set function with a physical interpretation. Evolving the equation $\phi(\mathbf{r},t) = 0$ in time defines the movement of the front. Differentiating with respect to time yields $\frac{\mathrm{d}}{\mathrm{d}t}\phi(\mathbf{r},t) = 0$ which is written as

$$\partial_t\phi(\mathbf{r},t) + \mathbf{V}\cdot\boldsymbol{\nabla}\phi(\mathbf{r},t) = 0, \tag{15}$$

where $\mathbf{V} = \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right|_{\mathbf{r}\in\Gamma}$ is the velocity of the zero level set.

Requiring the velocity field to be continuous leads to $\mathbf{V} = \mathbf{u}$, and the evolution equation for ϕ becomes

$$\phi_t + \mathbf{u}\cdot\boldsymbol{\nabla}\phi = 0. \tag{16}$$

IV. FEMLAB IMPLEMENTATION

One of the great advantages of the level set formulation is that it does not track the interface explicitly but rather capture it implicitly. Thereby we avoid to introduce explicit forces from the walls during collisions as they enter implicitly through the stress tensor $\boldsymbol{\sigma}$ and the no-slip boundary condition on the velocity field $\mathbf{u}$. Furthermore, several numerical tools are available for solving the dynamical system. In this section we describe how we have implemented the level set method in the finite element software package FEMLAB [7]. We have used the FEMLAB scripting language trough a MATLAB interface in the general PDE mode. Here the PDEs are given by

$$d_a\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \boldsymbol{\nabla}\cdot\boldsymbol{\Gamma} = \mathbf{F} \qquad \text{in } \Omega \tag{17a}$$

in terms of the variable vector $\mathbf{U}$, the current tensor $\boldsymbol{\Gamma}$ and the generalized source terfield $\mathbf{F}$. The boundary conditions take the form

$$-n_j\Gamma_{lj} = G_l + \frac{\partial R_m}{\partial U_l}\mu_m \qquad \text{on } \partial\Omega \tag{17b}$$

$$0 = R_m \qquad \text{on } \partial\Omega, \tag{17c}$$

where the index l is the variable counter, m is the constraint number (the number of boundaries) and j is the number space dimension number. The Lagrange multipliers μ_m are chosen by FEMLAB in order to fulfill the constraints, while the scalars F_l, G_l and R_m are given by the physics of the problem.

A. Navier–Stokes equation in Femlab

Introducing the characteristic length scale L_0, velocity scale U_0, density ρ_0, viscosity η_0 and surface tension γ_0 we can express the physical quantities as a dimensionless number times the characteristic scale. Denoting the nondimensional quantities by a tilde we simply have

$$\begin{aligned} \mathbf{r} = L_0\tilde{\mathbf{r}}, \quad \mathbf{u} = U_0\tilde{\mathbf{u}}, \quad \rho = \rho_0\tilde{\rho}, \\ \eta = \eta_0\tilde{\eta}, \quad \gamma = \gamma_0\tilde{\gamma}. \end{aligned} \tag{18}$$

Similarly we can define the characteristic pressure and timescale as relations between the chosen characteristic parameters

$$p = \frac{\eta_0 U_0}{L_0}\tilde{p}, \qquad t = \frac{L_0}{U_0}\tilde{t}. \tag{19}$$

Substituting Eqs. (18) and (19) into the Navier–Stokes equation (14) yields

$$Re\tilde{\rho}\left[\partial_{\tilde{t}}\tilde{\mathbf{u}} + (\tilde{\mathbf{u}}\cdot\tilde{\boldsymbol{\nabla}})\tilde{\mathbf{u}}\right] = \tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{\sigma}} + \frac{1}{Ca}\tilde{\gamma}\tilde{\kappa}\delta(\phi)\tilde{\boldsymbol{\nabla}}\phi. \tag{20}$$

Here the Reynolds number $Re = \rho_0 U_0 L_0/\eta_0$ is the ratio between inertial forces and viscous forces and the Capillary number $Ca = \eta_0 U_0/\gamma_0$ is the ratio between viscous forces and the surface tension forces.

Rearranging the terms in Eq. (20) we find

$$Re\tilde{\rho}\partial_{\tilde{t}}\tilde{\mathbf{u}} - \tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{\sigma}} = \frac{1}{Ca}\tilde{\gamma}\tilde{\kappa}\delta(\phi)\tilde{\boldsymbol{\nabla}}\phi - Re\tilde{\rho}(\tilde{\mathbf{u}}\cdot\tilde{\boldsymbol{\nabla}})\tilde{\mathbf{u}}, \tag{21}$$

which is seen to be on the Femlab general form if

$$d_a = Re\tilde{\rho}, \tag{22a}$$

$$\boldsymbol{\Gamma} = -\tilde{\boldsymbol{\sigma}}, \tag{22b}$$

$$\mathbf{F} = -Re\tilde{\rho}(\tilde{\mathbf{u}}\cdot\tilde{\boldsymbol{\nabla}})\tilde{\mathbf{u}} + \frac{1}{Ca}\tilde{\gamma}\tilde{\kappa}\delta(\phi)\tilde{\boldsymbol{\nabla}}\phi, \tag{22c}$$

$$\mathbf{U_u} = \tilde{\mathbf{u}}. \tag{22d}$$

The density $\tilde{\rho}$, viscosity $\tilde{\eta}$ and the curvature of the front $\tilde{\kappa}$ are defined as auxiliary functions of the level set function ϕ. In a system with two immiscible incompressible fluids (or a particle in a fluid) the density and viscosity are constant on each side of the interface. We can therefore define the dimensionless density and viscosity as

$$\tilde{\rho} = 1 + H(\phi)\left(\frac{\rho_1}{\rho_2} - 1\right) \tag{23}$$

and

$$\tilde{\eta} = 1 + H(\phi)\left(\frac{\eta_1}{\eta_2} - 1\right), \tag{24}$$

where $H(\phi)$ is a Heaviside function defined as

$$H(\phi) = \begin{cases} 1, & \phi \in \Omega_1, \\ 0, & \phi \in \Omega_2. \end{cases} \tag{25}$$

Setting $\rho_0 = \rho_2$ ensures that the density of the fluid is ρ_1 and ρ_2 in Ω_1 and Ω_2, respectively. Similarly setting $\eta_0 = \eta_2$ makes the viscosity of the fluid η_1 and η_2 in Ω_1 and Ω_2, respectively.

The curvature of the zero level set is given by

$$\kappa(\phi) = \boldsymbol{\nabla}\cdot\mathbf{n} = \boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\right), \tag{26}$$

where $\mathbf{n} = \boldsymbol{\nabla}\phi/|\boldsymbol{\nabla}\phi|$ is a unit normal vector to the interface [5, 8].

When solving the system numerically the abrupt change in density and viscosity across the interface causes numerical instabilities to occur. In order to avoid this we substitute $H(\phi)$, $\delta(\phi)$ and $\mathrm{sign}(\phi)$ with the smeared out versions $H_\epsilon(\phi)$, $\delta_\epsilon(\phi)$ and $\mathrm{sign}_\epsilon(\phi)$ defined as

$$H_\epsilon(\phi) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{\phi}{\epsilon}\right), \tag{27a}$$

$$\delta_\epsilon(\phi) = H'_\epsilon(\phi) = \frac{1}{2\epsilon} - \frac{1}{2\epsilon}\tanh^2\left(\frac{\phi}{\epsilon}\right), \tag{27b}$$

$$\mathrm{sign}_\epsilon(\phi) = \tanh\left(\frac{\phi}{\epsilon}\right). \tag{27c}$$

This implies that the interface has a finite thickness Γ_ϵ approximately given by

$$\Gamma_\epsilon = \frac{2\epsilon}{|\boldsymbol{\nabla}\phi|}. \tag{28}$$

B. The continuity equation in Femlab

The dimensionless form of the continuity equation is

$$0 = \tilde{\boldsymbol{\nabla}}\cdot\tilde{\mathbf{u}}, \tag{29}$$

which is entered into Femlab by choosing $\mathbf{F} = \tilde{\boldsymbol{\nabla}}\cdot\tilde{\mathbf{u}}$, $\boldsymbol{\Gamma} = \mathbf{0}$, $d_a = 0$ and $U_p = \tilde{p}$.

C. The level set equation in Femlab

The nondimensionalized form of the convection equation for the zero level set is

$$\phi_{\tilde{t}} + \tilde{\mathbf{u}}\cdot\tilde{\boldsymbol{\nabla}}\phi = 0, \tag{30}$$

which can be rearranged to

$$\phi_{\tilde{t}} = -\tilde{\mathbf{u}}\cdot\tilde{\boldsymbol{\nabla}}\phi \tag{31}$$

and implemented in Femlab by setting $\mathbf{F} = -\tilde{\mathbf{u}}\cdot\tilde{\boldsymbol{\nabla}}\phi$, $\boldsymbol{\Gamma} = \mathbf{0}$, $d_a = 1$ and $U_\phi = \tilde{\phi}$.

TABLE I: The parameter values used in the simulation of the test case.

Reynolds number	Re	$= 1 \times 10^{-3}$
Capillary number	Ca	$= 1 \times 10^{6}$
Density	ρ_0	$= 1 \times 10^{3}$ kg m^{-3}
Viscosity	η_0	$= 1 \times 10^{-1}$ Pa s
Obstacle size	l	$= 6 \times 10^{-6}$ m
Particle radius	r_p	$= 3 \times 10^{-6}$ m
Pressure drop	Δp	$= 1.2 \times 10^{-3}$ Pa
Time step	Δt	$= 5 \times 10^{-2}$ s
Mesh element size	`hmesh`	$= 1.1 \times 10^{-6}$ m
Thickness parameter	ϵ	$= 0.5 \times$ `hmesh`

D. Reinitialization of the level set function

It is necessary to maintain a uniform thickness of the interface throughout the calculations. This requires that the gradient of the level set function is constant within a region around the interface $|\phi| < \epsilon$. This is not automatically fulfilled. The time evolution of any level set $\phi(\mathbf{r}, t) = C$ is given by the level set Eq. (16). This means that the height of the level set function will remain constant, but it does not ensure that the gradient does not change. Thus in order to keep a fixed interface thickness we need to reinitialize the level set function without changing the zero level set.

In principle we can use any function that fulfills Eq. (6), since only the zero level set has a physical interpretation. But requiring the interface thickness to be fixed constrains the gradient of ϕ to be fixed in a region around the interface. A choice of $\phi(\mathbf{r}, t)$ that fulfills these requirements is the signed distance function, where the distance is the shortest distance $d(\mathbf{r})$ from a point to the interface

$$d(\mathbf{r}) = \pm \min(|\mathbf{r} - \mathbf{r}_\Gamma|), \qquad (32)$$

$\mathbf{r}_\Gamma$ being the points on the interface. The plus sign applies if $\mathbf{r} \in \Omega_1$ and the minus sign if $\mathbf{r} \in \Omega_2$. The length of the gradient for this particular choice of level set function is

$$|\boldsymbol{\nabla}\phi| = 1. \qquad (33)$$

We have implemented two different reinitilization procedures. One simple reinitialization procedure where we recalculate the level set function at every time step and one using the reinitialization equation suggested by Sussmann, Smereka and Osher [9]

$$\partial_\tau \psi(\mathbf{r}, \tau) = \mathrm{sign}(\phi)\big(1 - |\boldsymbol{\nabla}\psi(\mathbf{r}, \tau)|\big), \qquad (34)$$

with the initial condition $\psi(\mathbf{r}, 0) = \phi$ and τ being a pseudotime. The steady state solution to this equation is the reinitialized level set function. Because numerical oscillations can occur if the sign of ϕ changes abruptly at the interface it is necessary to use the smeared out sign function given in Eq. (27c).

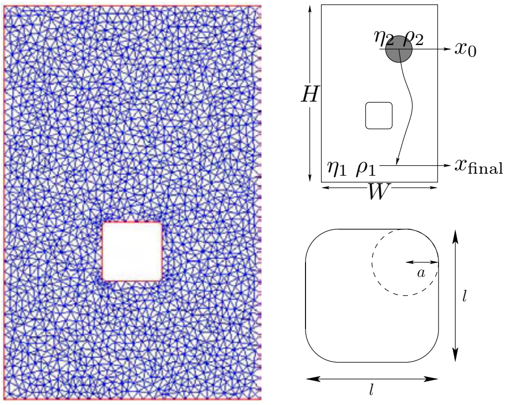


FIG. 1: For the test study we use the geometry and mesh shown in the figure. The general shape of the obstacle is as shown in the lower inset on the right. The radius a of the rounded corner was changed from one simulation to the next. The aspect size of the obstacle is l. The height of the channel is $H = (20/3)l$ and the width of the channel is $W = (13/3)l$. The upper inset on the right shows the general idea of the test study: The particles start in the initial position x_0 and the final position x_{final} is recorded.

The reinitialization equation is already on a form suitable for implementation in FEMLAB. Simply letting F equal the right hand side of the equation and setting $d_a = 1$ and $\mathbf{\Gamma} = \mathbf{0}$ with $U_\psi = \psi$ does the trick.

To avoid mass loss during the reinitialisation procedure we have put a constraint on the solution: the volume of the particle must be constant at all time. This is done in FEMLAB via the field `fem.equ.constr` where we constrain the difference between the integrals of the smeared out Heaviside function $H_\epsilon(\psi)$ at time τ and the smeared out Heaviside function $H_\epsilon(\phi)$ at time $t = 0$ to be zero. The integrals are computed by using the integration coupling variables in FEMLAB.

V. MODEL SYSTEM AND SETUP

To test the implementation of the level set method in FEMLAB we have done a test study of a particle (a drop of high viscosity and surface tension) which is passively convected in a two dimensional fluid flow. The viscosity η_2 of the particle was 100 times larger than the viscosity η_1 of the fluid. The density ρ_1 of the fluid was equal to the density ρ_2 of the particle. The complete list of parameters is given in Table I.

The physical domain is an infinitely wide and infinitely long channel with an obstacle in the center as shown in Fig. 1. The boundary conditions on the fluid are no-stress on the sides of the computational domain and no-slip at the obstacle. The fluid velocity field is periodic from

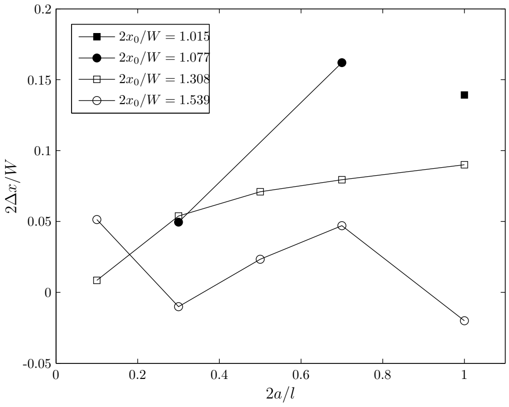


FIG. 2: For particles passing obstacles of different shapes normalized difference $2\Delta x/W$ in horizontal position from start to finish is plotted versus starting position $2a/l$. The missing data points for the simulations with the initial positions of the particles nearest to the center of the channel is due to the particles getting stuck at the obstacle and hence not reaching the final position.

top to bottom of the domain and is driven by a pressure difference Δp.

We ran a series of simulations with the shape of the obstacle changing from circular to quadratic by changing the radius of the rounded obstacle corner a. Each simulation consisted of a series of runs with different initial horizontal position x_0 of the particles and the initial vertical position of the particles was $y_0 = H - l$ from the top of the channel. When the center of a convected particle is l from the bottom of the channel the final horizontal position $x_{\rm final}$ is detected (Fig. 1).

We represent the particle by the negative part of a level set function and the surrounding fluid is identified by the positive part of the level set function. The initial level set function is given by

$$\phi(x, y, t = 0) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r_p, \qquad (35)$$

where (x_0, y_0) is the initial position of the particle and r_p is the radius of the particle. Using these parameters we solve the problem by first evolving the dynamical equations in a small time step Δt and then reinitialize the level set function using the reinitialization procedures described above. With the reinitialized level set function as initial condition for ϕ we evolve the dynamical system one more time step. This sequence is continued until the particle has moved all the way through the system.

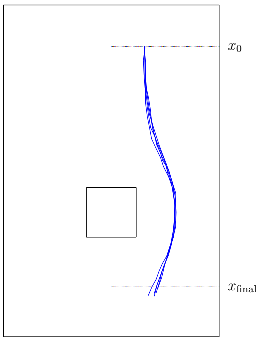


FIG. 3: The paths of particles passing obstacles of different shapes when the starting point is $2x_0/W = 0.308$ right of the centerline of the channel.

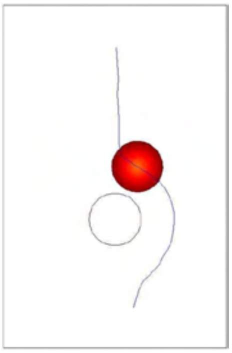


FIG. 4: The path of the particle started at $2x_0/W = 0.015$ when the radius of the rounded obstacle corner is $a = l/2$. The particle (black dot) is shown when it 'interacts' with the obstacle. The small gap between the particle and the obstacle wall is caused by the smearing of the particle interface.

VI. RESULTS

We carried out simulations for four different initial positions of the particle. The initial horizontal positions $2x_0/W$ were 0.015, 0.077, 0.308 and 0.539, respectively. For each of these initial positions we used five different radii of the rounded corner of the obstacle: $2a/l = i/10$, with $i = 1, 3, 5, 7, 10$.

For each combination of initial position and obstacle shape we solved the system and obtained the particle paths. Examples are shown in Figs. 3 and 4. It is seen that the paths of particles with the same initial position changes as function of the shape of the obstacle (Fig. 3). In Fig. 2 we have plotted the difference in the horizontal position Δx from start to finish.

The difference in horizontal position is almost zero for the particles started in at the greatest distance from the center of the channel, independent of the shape of the obstacle. As the initial position gets closer to the center of the channel the difference in horizontal position becomes larger and the round obstacles tend not to drag as much in the particles as the square obstacles yielding a larger difference in the horizontal position.

Fig. 4 shows that our implementation of the level set method is capable of coping with the interaction forces between the stable obstacles and the moving particles automatically.

VII. DISCUSSION AND CONCLUSIONS

We have shown that the level set method is easily implementable in FEMLAB and that it is a suitable method for coping with the interaction forces between particles and hard walls automatically. Particles can be modelled as very viscous liquid drops and the shape preservation can be taken care of trough an appropriate reinitialization procedure.

We have used a simple shape preserving reinitialization method. Further work is needed in order to convect particles of an arbitrary fixed shape. One promising reinitialisation scheme is the particle level set method suggested by Enright *et al.* [10].

The level set method might prove useful when simulating microfluidic systems for particle handling. In this paper we have only considered the forces exerted on the particles by the convecting fluid and thereby indirectly the forces from the solid walls. However also other forces such as DEP forces or magnetic forces could be taken into account making the method applicable for simulations of many lab-on-a-chip systems fabricated today.

APPENDIX A

We demonstrate how to construct the transverse level set function ψ with the required properties. We start by defining a coordinate transformation by

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\Big(x(s,\tau),y(s,\tau)\Big) = \boldsymbol{\nabla}\phi\Big(x(s,\tau),y(s,\tau)\Big), \qquad \text{(A1a)}$$

where

$$\Big(x(s,0),y(s,0)\Big) = \Big(\overline{x}(s),\overline{y}(s)\Big). \qquad \text{(A1b)}$$

Because of the δ function in Eq. (14) ψ only needs to fulfill the requirements in a small region $|\tau| < \epsilon$ around Γ. In this small region we can define ψ as

$$\psi\Big(x(s,\tau),y(s,\tau)\Big) = \psi_0(s), \qquad \text{(A2)}$$

where $\psi_0(s)$ is a smooth increasing function if and only if the mapping of (x,y) to (s,τ) is one-to-one. Using the change of variables theorem we have to show that

$$\left|\frac{\partial(x,y)}{\partial(s,\tau)}\right| \neq 0. \qquad \text{(A3)}$$

Taylor expanding Eq. (A1a) around $\tau = 0$ yields

$$(x_\tau, y_\tau) = \boldsymbol{\nabla}\phi\Big(\overline{x}(s),\overline{y}(s)\Big) + \mathcal{O}(\tau). \qquad \text{(A4)}$$

Differentiation of Eq. (A1a) with respect to s and integration with respect to τ yields

$$\int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\frac{\mathrm{d}}{\mathrm{d}\tau'}\Big(x(s,\tau'),y(s,\tau')\Big)\,\mathrm{d}\tau' = \int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\nabla}\phi\Big(x(s,\tau'),y(s,\tau')\Big)\,\mathrm{d}\tau' \qquad \text{(A5)}$$

From which follows

$$\Big(x_s(s,\tau),y(s,\tau)\Big) - \Big(x_s(s,0),y_s(s,0)\Big) = \int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\nabla}\phi\Big(x(s,\tau'),y(s,\tau')\Big)\,\mathrm{d}\tau', \qquad \text{(A6)}$$

and thus

$$\begin{aligned}&\Big(x_s(s,\tau),y(s,\tau)\Big)\\ &= \Big(\overline{x}_s(s),\overline{y}_s(s)\Big) + \int_0^\tau \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{\nabla}\phi\Big(x(s,\tau'),y(s,\tau')\Big)\,\mathrm{d}\tau'\\ &= \mathbf{T}(s) + \mathcal{O}(\tau). \qquad \text{(A7)}\end{aligned}$$

Here $\mathbf{T}$ is a unit tangent vector to the interface. We can now calculate the determinant (A3)

$$\begin{aligned}\left|\frac{\partial(x,y)}{\partial(s,\tau)}\right| &= (x_\tau,y_\tau)\cdot(-y_s,x_\tau)\\ &= \boldsymbol{\nabla}\phi(\overline{x}_s,\overline{y}_s)\cdot\hat{\mathbf{T}} \qquad \text{(A8)}\\ &= |\boldsymbol{\nabla}\phi||\mathbf{T}| + \mathcal{O}(\tau)\\ &= |\boldsymbol{\nabla}\phi|_{\phi=0} + \mathcal{O}(\tau) \neq 0.\end{aligned}$$

This means that ψ is well defined in a small region around Γ. Now all we need to prove is that $\boldsymbol{\nabla}\phi$ and $\boldsymbol{\nabla}\psi$ are orthogonal and that $|\boldsymbol{\nabla}\psi| \neq 0$. The orthogonality can be proved by differentiating ψ with respect to τ,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\psi\Big(x(s,\tau),y(s,\tau)\Big) &= \psi_x x_\tau + \psi_y y_\tau\\ &= \boldsymbol{\nabla}\psi\cdot\boldsymbol{\nabla}\phi = \frac{\mathrm{d}\psi_0(s)}{\mathrm{d}\tau} = 0,\end{aligned} \qquad \text{(A9)}$$

which means that ϕ and ψ are orthogonal if and only if $|\boldsymbol{\nabla}\psi| \neq 0$. This follows immediately from differentiating ψ with respect to s,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}s}\psi\Big(x(s,\tau),y(s,\tau)\Big) &= \psi_x x_s + \psi_y y_s\\ &= \boldsymbol{\nabla}\psi\cdot(x_s,y_s) \qquad \text{(A10)}\\ &= \boldsymbol{\nabla}\psi\cdot\mathbf{T}\\ &= |\boldsymbol{\nabla}\psi| = \psi_0'(s) > 0,\end{aligned}$$

because $\psi_0(s)$ was chosen to be an increasing function. Thereby we have established the level set formulation of the Navier–Stokes equation for a two liquid flow of incompressible fluids.

[1] C.-F. Chou, O. Bakajin, S. W. P. Turner, T. A. J. Duke, S. S. Chan, E. C. Cox, H. G. Craighead, and R. H. Austin, USA **96**, 13762 (1999).

[2] T. A. J. Duke and R. H. Austin, Phys. Rev. Lett. **80**, 1552 (1998).

[3] L. R. Huang, E. C. Cox, R. H. Austin, and J. C. Sturm, Science **304**, 987 (2004).

[4] L. R. Huang, P. Silberzan, J. Tegenfeldt, E. C. Cox, J. C. Sturm, R. H. Austin, and H. Craighead, Phys. Rev. Lett. **89**, 1 (2002).

[5] J. A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge University Press, 1999), 2nd ed.

[6] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, J. Comput. Phys. **124**, 449 (1996).

[7] FEMLAB homepage, `www.comsol.dk`.

[8] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153 of *Applied mathematical sciences* (Springer-Verlag New York, 2003), 1st ed.

[9] M. Sussman and E. Fatemi, SIAM J. Sci. Comput. **20**, 1165 (1999).

[10] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell (2002).

Appendix B

Poster

Title

The level set method for interface dynamics in microfluidics.

Authors

Martin Heller and Henrik Bruus.

Conference

Presented at the 6$^{\text{th}}$ Liquid Matter Conference in Utrecht (NL), July 2–6, 2005.

Abstract

A major challenge in theoretical microfluidics is the study of dynamics and evolution of interfaces. One example is the interaction between a suspended particle and the walls of the microchannel containing it; another example is the topology changes in connection with the merge of two bubbles. The level set method is well suited to cope with these problems. By introducing a hypersurface $\phi(\mathbf{r}, t)$, the particle/bubble interface is represented as the zero level set $\phi(\mathbf{r}, t) = 0$. The major advantage of the method is that this zero level set can be calculated implicitly instead of explicit tracking of the points on the interface.
The level set function is initialized as the distance to the interface and evolves according to

$$\phi_t = -\mathbf{u} \cdot \boldsymbol{\nabla}\phi.$$

This coupled with the incompressibility condition and the Navier-Stokes equations

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \quad \text{and} \quad \rho\mathbf{u}_t + \rho(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{f},$$

describes the problem completely. Here $\mathbf{u}$ is the velocity field of the fluid, ρ is the density, $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{f}$ is the forces acting on the interface.

We have implemented the level set method in the numerical simulation tool FEMLAB along with a re-initialization procedure for the level set function as well as a volume conserving constraint.

As a test case we have studied the motion of a non-spherical microparticle convected by a viscous liquid in a nontrivial channel geometry. In particular we have studied the translation and rotation of the particle as well as its interaction with the walls, all induced by the stress field of the fluid. Our implementation is useful when modeling particle and cell-handling in lab-on-a-chip systems.

The level set method for interface dynamics in microfluidics

Martin Heller and Henrik Bruus

MIC – Department of Micro and Nanotechnology, DTU bldg. 345 east
Technical University of Denmark
DK–2800 Kgs. Lyngby, Denmark

Motivation

In recent years many new microfluidic devices have been developed for separation and sorting of particles and DNA in aqueous solutions. A special class of separation devices are the bumper arrays. Here mechanical interaction between the convected particles and rigid obstacles in the microfluidic channels makes fast deterministic separation possible on the microscale. Using a Femlab implementation of the level set method, we have studied the motion of finite sized particles in microfluidic channels with obstacles. The method implicitly copes with the interaction forces between the suspended particles and the walls of the microchannel. As a test case we have studied the motion of a spherical microparticle convected by a viscous liquid in a nontrivial channel geometry.

The algorithm of the level set method

The level set function $\phi(\mathbf{r},t)$ is initialized as a signed distance function

$$\phi(\mathbf{r},0)=\sqrt{(x-x_0)^2-(y-y_0)^2}-R$$

The level set function is evolved one time step according to the equation

$$0=\phi_t+\mathbf{u}\cdot\nabla\phi$$

The level set function is reinitialized

$$\psi_\tau=\mathrm{sign}(\phi_0)\left(1-|\nabla\psi|\right) \qquad ,\phi(\mathbf{r},0)=\phi_0$$

The Navier–Stokes equation is solved

$$\rho\left[\partial_t\mathbf{u}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right]=\nabla\cdot\mathbf{ó}+\gamma\kappa\,\mathbf{n}$$

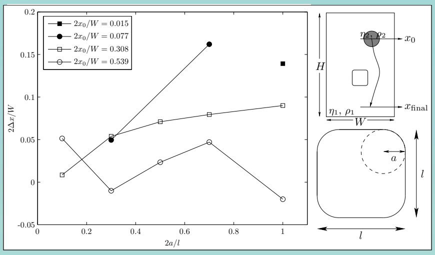
using no-slip boundary conditions and the incompressibility condition

$$\nabla\cdot\mathbf{u}=0$$

Notation:

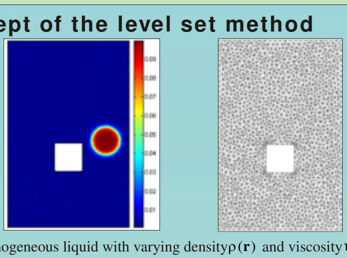
$\mathbf{r}=(x,y)$: Position vector	$\mathbf{u}$: Velocity field
R : Radius of the particle	$\mathbf{ó}$: Stress tensor
ϕ : Level set function	$\tilde{a}$: Surface tension
φ : Reinitialized level set function	κ : Curvature of the particle surface
t,τ : Time scales	i.e. where $\phi(\mathbf{r})$ is zero.

Numerical results from test run



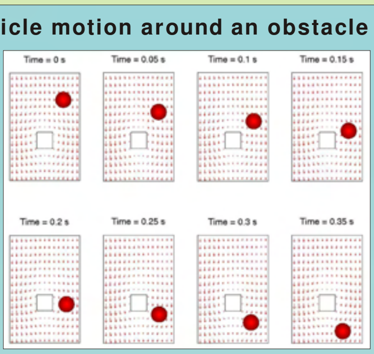
The normalized horizontal displacement plotted versus obstacle shape for particles starting at different positions.

Concept of the level set method



One inhomogeneous liquid with varying density $\rho(\mathbf{r})$ and viscosity $\eta(\mathbf{r})$.
The particle is represented by the high viscosity region (see left figure).
The geometry of this region is defined by the level set function $\phi(\mathbf{r})=0$.
This enables computation of the time dependent solution using a fixed mesh.

Particle motion around an obstacle



Velocity field (arrows) and the convected particle (red circle).
The volume of the particle is fixed by a constraint on the solution,
while the shape is preserved by the surface tension.

Conclusions and outlook

We have demonstrated that the level set method can be used to simulate motion of finite sized particles in microchnnels with obstacles.

The advantage of the method is that interaction forces between the particle and the rigid walls are taken into account implicitly through the surface tension and the no-slip velocity condition on the obstacles.

The drawback of the method is the long computation time caused by the volume conserving constraint imposed on the solution. Future work therefore involves finding a more efficient reinitialization procedure.

More info: www.mic.dtu.dk/MIFTS

Appendix C

Code examples

C.1 Geometry and equation system

```
% 1. Geometry definition
G   = rect2(0,W,0,H);
r   = rect2(1e-5,1.6e-5,1.2e-5,1.8e-5);
b1  = fillet(r,'radii',0.29e-5*j/10);
[geom,ctx,ptx] = geomcsg({G b1},{},'ns',{'a' 'b'},'sf','a-b','out',{'g','ctx','ptx'});
fem.geom = geom;

fem.mesh = meshinit(fem,'hauto',6,'hmax',hmesh,'hcurve',4,'report','off');
fem.xmesh = meshextend(fem);

% Defining the equation system
% Space coordinates
fem.sdim = {'x' 'y'};

% Variables
fem.dim = {'u' 'v' 'p','phi','psi'};
fem.shape  = {shlag(2,'u'),shlag(2,'v'),shlag(1,'p'),shlag(2,'phi'),shlag(2,'psi')};
V0 = real(postint(fem,'1-(0.5+0.5*tanh((min(min(sqrt((x-xp)^2+(y-yp)^2)-rp,sqrt((x-xp)^2+(
     y-(yp+H))^2)-rp),sqrt((x-xp)^2+(y-(yp-H))^2)-rp)/(alpha*hmesh))))'));

% Constants and expressions
fem.const = [fem.const,{'V0',V0}];
fem.expr = {'epsilon','alpha*hmesh',...
            'sigmaxx','2*eta*ux-p',...
            'sigmaxy','eta*(vx+uy)',...
            'sigmayx','sigmaxy',...
            'sigmayy','2*eta*vy-p',...
            'kappa','(phixx*phiy^2-2*phix*phiy*phixy+phiyy*phix^2)/(phix^2+phiy^2)^1.5'
                 ,...
            'rho','rho0',...
            'eta','eta0*(1-(1-et)*(1-(1+tanh(phi/epsilon))/2))',...
            'U','sqrt(u^2+v^2)',...
            'grad_phi','sqrt(phix^2+phiy^2)',...
            'grad_psi','sqrt(psix^2+psiy^2)',...
            'phi_init','sqrt((x-xp)^2+(y-yp)^2)-rp'};

% Governing equations
fem.equ.ind = [1];
fem.equ.shape = {[1:5]};
fem.form = 'general';
fem.equ.da = {{'Re*ro','0','0','0','0';...
               '0','Re*ro','0','0','0';...
               '0','0','0','0','0';...
               '0','0','0','1','0';...
               '0','0','0','0','1'}};
fem.equ.f =  {{'-Re*ro*(u*ux+v*uy)+1/(Ca)*gamma*kappa*(1-tanh(phi/epsilon)^2)/(2*epsilon)*
     phix';...
               '-Re*ro*(u*vx+v*vy)+1/(Ca)*gamma*kappa*(1-tanh(phi/epsilon)^2)/(2*epsilon)*
                    phiy';...
               'ux+vy';...
```

```
                  '-u*phix';'-v*phiy';...
                  'tanh(phi/epsilon)*(1-grad_psi)'}};
fem.equ.ga = {{{'-sigmaxx';'-sigmayx'};...
               {'-sigmaxy';'-sigmayy'};...
               {'0';'0'};...
               {'0';'0'};...
               {'0';'0'}}};
fem.equ.constr = {{0 0 0 'V_phi-V0' 'V_psi-V0'}};
```

Femlab code 3: Code for setting up geometry and equation system in

C.2 Initialization of the level set function

```
init = asseminit(fem,'init',{'u' 'u' 'v' 'v' 'p' 'p' 'phi' 'psi' 'psi' 'psi'},'u',fem0.sol
    ,'solnum',[n]);
```

Femlab code 4: Initial value for the level set function is the reinitialized level set function.

C.3 Solution of the physical problem

```
fem.sol = femtime(fem, ...
                  'init',init, ...
                  'nullfun','flspnull',...
                  'solcomp',{'u' 'v' 'p' 'phi'},...
                  'outcomp',{'u' 'v' 'p' 'phi'},...
                  'tlist',[0:0.01:0.05], ...
                  'linsolver','gmres',...
                  'tout','tlist', ...
                  'report','off');
fem0 = fem;
```

Femlab code 5: Solver command for the convection equation.

C.4 Reinitialization

```
init = asseminit(fem,'init',{'u' 'u' 'v' 'v' 'p' 'p' 'phi' 'phi' 'psi' 'phi'},'u',fem0.sol
    ,'solnum',[n]);

fem.sol=femtime(fem, ...
                'nullfun','flspnull',...
                'init',init, ...
                'solcomp',{'psi'}, ...
                'outcomp',{'psi','phi','u','p','v'}, ...
                'tlist',[0:0.01:0.3], ...
                'tout','tlist');
```

Femlab code 6: Solver command for the reinitialization equation. The initial during reinitialization is the level set function. We only solve for the reinitialized level set function but save the solution values of the physical problem.

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1999.

[2] Comsol A/S. Femlab homepage. `www.comsol.dk`.

[3] Comsol A/S. Femlab *3 Modeling Guide*, January 2004.

[4] G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, first cambridge mathematical library edition, 2002.

[5] Henrik Bruus. *Theoretical Microfluidics*. DTU lecture notes, Microfluidics Theory and Simulation, 2004. Aveilabel on-line at `http://www.mic.dtu.dk/research/MIFTS/publications/books/MIFTSnote.pdf`.

[6] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124(2):449–64, 1996.

[7] D. L. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106(1):77–91, 1993.

[8] Chia-Fu Chou, Olgica Bakajin, Stephen W. P. Turner, Thomas A. J. Duke, Shirley S. Chan, Edward C. Cox, Harold G. Craighead, and Robert H. Austin. Sorting by diffusion: An asymmetric obstacle course for continuous molecular separation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(24):13762–13765, 1999.

[9] T. A. J. Duke and R. H. Austin. Microfabricated sieve for the continuous sorting of macromolecules. *Physical Review Letters*, 80(7):1552–5, 1998.

[10] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[11] Richard P. Feynman. There's plenty of room at the bottom. *Journal of Microelectromechanical Systems*, 1(1):60–66, 1992.

[12] ISO Standards Handbook. *Quantities and units*. International Organization for Standardization, 3 edition, 1993. ISBN: 92-67-10185-4.

[13] Lotien Richard Huang, Edward C. Cox, Robert H. Austin, and James C. Sturm. Tilted brownian ratchet for DNA analysis. *Analytical Chemistry*, 75(24):6963–6967, 2003.

[14] Lotien Richard Huang, Edward C. Cox, Robert H. Austin, and James C. Sturm. Continuous particle separation through deterministic lateral displacement. *Science*, 304:987–990, 2004.

[15] Lotien Richard Huang, Pascal Silberzan, Jonas O. Tegenfeldt, Edward C. Cox, James C. Sturm, Robert H. Austin, and Harold Craighead. Role of molecular size in rachet fractionation. *Physical Review Letters*, 89:178301–1–178301–4, 2002.

[16] George Em Karniadakis and Ali Beskok. *Micro Flows. Fundamentals and Simulation.* Springer, 2000.

[17] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics*, volume 6 of *Course of theoretical physics.* Elsevier Science Limited, 2 edition, 2003.

[18] B. Lautrup. *Physics of continous matter: exotic and everyday phenomena in the macroscopic world.* Institute of Physics Publishing, 2005. ISBN: 0750307528.

[19] Sune Lomholt. *Numerical investigations of macroscopic particls dynamics in microflows.* PhD thesis, Risø National Laboratory, Roskilde, Denmark, 2001.

[20] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied mathematical sciences.* Springer, Springer-Verlag New York Inc., 1 edition, 2003. ISBN: 0-387-95482-1.

[21] Stephen D. Senturia. *Microsystem Design.* Kluwer Academic Publishers Group, 4 edition, 2000. ISBN: 0-7923-7246-8.

[22] J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 2 edition, 1999. ISBN: 0-521-64557-3 paperback.

[23] Mark Sussman and Emad Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.

[24] S. Turek and M. Schäfer. Benchmark computations of laminar flow around a cylinder. *Flow Simulation with High-Performance Computers*, II:547–566, 1996.

[25] B H Weigl and P Yager. MICROFLUIDICS: Microfluidic diffusion-based separation and detection. *Science - AAAS - Weekly Paper Edition*, 283(5400):346–347, 1999.