Two Phase Flow by Level Set Method

Rune Barnkob and Mathias Bækbo Andersen

Supervisors: Professor Henrik Bruus and PhD Student Martin Heller

MIC – Department of Micro and Nanotechnology
Technical University of Denmark

17 January 2007
Preface

This report is written at the Technological University of Denmark. The Department of Micro and Nanotechnology offers the three week course 33443 Microfluidics Theory and Simulation, which sorts under the Microfluidics Theory and Simulation (MIFTS) group [4], headed by Professor Henrik Bruus.

The motivation for the present report is built on a contact from the Swedish company HemoCue [3], who requests theoretical calculations on their blood analysis products. Development of the HemoCue products has until now based on a trial and error approach and further progress now demands a more theoretical and analytical treatment. The main problem is air bubbles that get stuck in the central analysis chamber, which is filled using the capillary forces. Optical measurements on the blood are very sensitive to these bubbles that give rise to incorrect outcomes. It is therefore decided that this report shall act as a preliminary treatment of the problem.

During the three weeks the authors and their supervisors has visited the R&D department of HemoCue in Angelholm, Sweden. It has been an interesting meeting, which first of all has given an introduction to the company and their visions, but has also clarified the specific problem of trapped bubbles. This report focuses on following a two phase flow interface by using a level set method, which can simulate complex flows. It does not give the specific answer for the HemoCue problem, but provides a tool, which together with analytical treatments of the problem, can provide HemoCue with the necessary guidelines when developing new micro cuvettes.

Throughout the report the Einstein summation rule will be used when appropriate. The reader is assumed to be familiar with this notation. For notational simplicity \( \partial_x \) will be used instead of \( \frac{\partial}{\partial x} \) etc.

We hope the reader will enjoy reading the report.

Rune Barnkob and Mathias Bækbo Andersen
MIC – Department of Micro and Nanotechnology
Technical University of Denmark
17 January 2007
Contents

1 Theory .......................... 1
  1.1 Capillary Effects ..................... 1
  1.2 Analytical analysis of capillary micro pump .................... 3
  1.3 A conservative level set method for two phase flow .................... 5

2 Numerical implementation .................. 9
  2.1 Setting up the model .................... 9
    2.1.1 Domain variables .................... 11
    2.1.2 Boundary variables .................... 12

3 Results .................................. 15

4 Conclusions .................................. 19
Chapter 1

Theory

In this chapter the general physics that give rise to capillary effects is presented. Physical quantities such as surface tension and contact angle are described. The length scale for which capillary effects are dominant is found. This will explain why capillary rise is not observed in, e.g., an ordinary water bottle but is unavoidable in small structures such as the HemoCue micro cuvettes. Finally, the theory for the level set method is briefly reviewed. The level set method makes complicated simulations of capillary effects possible and can be a relevant tool, when designing new products with optimal functionality.

1.1 Capillary Effects

As things get smaller the surface to bulk ratio increases and at a certain characteristic length, surface forces dominate the system and so-called capillary effects appear. For a blood-air interface this characteristic capillary length is given by \( l_{\text{cap}}^{\text{blood-air}} \approx 2.4 \text{ mm} \). A very used capillary effect is the capillary pump, where a liquid is sucked into a narrow channel due to surface forces.

An important concept, when considering capillary effects, is the surface tension. Since it costs energy to create a surface, the surface tension, \( \gamma \), of an interface is defined as the Gibbs free energy per surface area. The surface tension for an blood-air interface is approximately 60 mJ m\(^{-2}\). When having a surface tension the interface will curve and thus give rise to the so-called Young-Laplace pressure drop, \( \Delta p \), across the curved interface

\[
\Delta p = \left( \frac{1}{R_1} + \frac{1}{R_2} \right) \gamma, \tag{1.1}
\]

where \( R_1 \) and \( R_2 \) are the radii of curvature. An everyday analog of the Young-Laplace pressure drop is blowing up of a balloon. Here one obviously adds pressure to the internal volume of the balloon, which gives rise to a surface tension and radii of curvature.

Another important parameter is the contact angle, \( \theta \), which appears at the contact line between three different phases, which is typically two immiscible fluids and a solid wall. The contact angle is an important parameter, since it is directly related to the radii of curvature and the surface tensions between the different phases. This is given by Young’s
Figure 1.1: Graphical representation of the contact angle, $\theta_c$. The figure shows a drop of liquid on a substrate. There are three interfaces all denoted with the corresponding surface tension, $\gamma_{sl}$, $\gamma_{lg}$ and $\gamma_{sg}$. The contact line is the geometrical line where the three phases meet and the contact angle is measured between the tangents to the solid-liquid and liquid-gas interfaces at this line. The picture is from WikiPedia.

\[
\cos \theta = \frac{\gamma_{sg} - \gamma_{sl}}{\gamma_{lg}},
\]

where s, g and l refer to the solid, gas and liquid phases, respectively. See Fig. 1.1. This section is build on [1].
1.2 Analytical analysis of capillary micro pump

Consider a micro channel consisting of two infinite parallel plates separated by the height \( h \), see Fig. 1.2. The infinite parallel plate model is a good approximation for a wide channel with a low height to width ratio, which is the case when analysing the HemoCue micro cuvette. As usual in micro fluidics no-slip boundary conditions is assumed at the walls

\[
\mathbf{v}(\mathbf{r}) = \mathbf{0}, \quad \text{for } \mathbf{r} \in \partial \Omega. \tag{1.3}
\]

Due to translational invariance in the \( x \)- and \( y \)-direction the velocity field only depends on \( z \). If the driving force is directed along the \( x \)-axis, only the \( x \)-component of the velocity field is non-zero

\[
\mathbf{v}(\mathbf{r}) = v_x(z) \mathbf{e}_x. \tag{1.4}
\]

This form of the velocity field renders the Navier-Stokes equation linear. In the quasistationary regime the velocity is independent of time and Navier-Stokes equation is simplified to an ordinary differential equation of second order. Assuming, for simplicity, that the infinite parallel plate channel is oriented perpendicular to the field of gravity, one can justify that the force of gravity is cancelled out due to hydrostatic pressure in the \( z \)-direction. The curvature of the meniscus gives rise to a finite pressure difference across the interface as described in the Young-Laplace equation, Eq. (1.1). Assuming both ends of the channel are exposed to the same pressure, e.g., atmospheric pressure, the fluid will experience a pressure difference, \( \Delta p \), when going from \( x = 0 \) to the position of the meniscus, \( L(t) \). This pressure difference is assumed to be linearly dependent on the position. Under the above mentioned circumstances one can describe the flow of the fluid by a Poiseuille flow if the small deviations close to the meniscus are neglected. Furthermore one has to assume that the timescale for the advancement of the meniscus is sufficiently long, such that the Poiseuille flow is fully developed. From [1] it is known that for a typical microfluidic system it takes approximately 2 ms for a Poiseuille flow to be fully developed. As described in [1, p. 100] one can, by use of mass conservation and incompressibility, identify the velocity of the moving meniscus with the average velocity of the Poiseuille flow

\[
\frac{dL(t)}{dt} = V_0 = \frac{Q}{A}, \tag{1.5}
\]
where $L$ is the position of the meniscus, $Q$ is the volume flow rate and $A$ is the cross sectional area of the rectangular channel. The volume flow rate for a Poiseuille flow in an infinite parallel plate channel is calculated by use of the Hagen-Poiseuille law,

$$Q = \frac{\Delta p}{R_{hyd}},$$  \hspace{1cm} (1.6)

where $R_{hyd}$ is the hydraulic resistance of the channel. The hydraulic resistance of an infinite parallel plate channel is given by

$$R_{hyd} = 12\eta L(t) \frac{1}{h^3 w},$$ \hspace{1cm} (1.7)

with $\eta$ being the viscosity of the fluid\(^1\) and $w$ being the width of a finite section in the $y$-direction. Eq. (1.5) becomes an ordinary first order differential equation for the position of the meniscus. The solution is stated in [1, Eq. (5.34)] and reproduced below

$$L(t) = h \sqrt{\frac{t}{\tau_{adv}}}, \hspace{1cm} \tau_{adv} \equiv \frac{3\eta h}{\gamma \cos(\theta)},$$  \hspace{1cm} (1.8)

Note that the position goes as the square root of time and is proportional to the square root of the height, $\sqrt{h}$, which means that the fluid propagates faster as the height, $h$, of the channel increases. If one chooses a characteristic length and time, e.g. the height of the channel and $\tau_{adv}$, Eq. (1.8) can be non-dimensionalized

$$\tilde{L} = \frac{L}{h},$$  \hspace{1cm} (1.9)

$$\tilde{t} = \frac{t}{\tau_{adv}},$$  \hspace{1cm} (1.10)

Eq. (1.8) then reads

$$\tilde{L}(\tilde{t}) = \sqrt{\tilde{t}}.$$  \hspace{1cm} (1.11)

This simple result for the position of the meniscus is compared to the results from the numerical simulation in Chapter 2 in order to validate the COMSOL code.

---

\(^1\)Whole blood at 37 °C and 1 atm has a viscosity of $\eta_{\text{blood}} = 3.26 \pm 0.43 \times 10^{-3}$ Pa s (http://www.clinchem.org/cgi/content/abstract/42/8/1189)
1.3 A conservative level set method for two phase flow

The level set methods were first introduced in 1988 by Osher and Sethian. It is a class of numerical techniques that deal with fluid-interface motion. The interface is represented implicitly and the equation of motion is numerically approximated using schemes built from hyperbolic-conservation laws. Level set methods are particularly useful for problems in which the topology of the evolving interface changes during the course of events and for problems in which sharp corners and cusps are present.

Consider a simple closed curved curve \( I(t) \) moving in two dimensions. A given velocity field, \( \mathbf{v} = (v_x, v_y) \), transports the interface. The so-called analysis-view considers the function \( \phi : \mathbb{R}^2 \times [0, \infty) \rightarrow \mathbb{R} \) where the zero-level set \( \phi = 0 \) corresponds to the evolving front \( I(t) \). This defines the front implicitly. The level set function is constricted to a domain

\[
\phi = \phi(\mathbf{r}, t), \quad \mathbf{r} \in \Omega, \tag{1.12}
\]

with \( \Omega \) being the domain of interest, defined by the physical problem. The equation for the evolution of \( \phi \) corresponding to the motion of the interface is given by the convection equation

\[
\partial_t \phi + \mathbf{v} \cdot \nabla \phi = 0. \tag{1.13}
\]

If the flow is incompressible the velocity field will be divergence free, \( \nabla \cdot \mathbf{v} = 0 \). This fact can be shown using the continuity equation. Eq. (1.13) can then be written as

\[
\partial_t \phi + \nabla \cdot (\mathbf{v} \phi) = 0, \tag{1.14}
\]

which corresponds to a continuity equation of the level set function. The numerical approach approximates the solution to the time-dependent initial value problem to follow the evolution of the associated level set function, whose zero-level set always gives the location of the propagating interface, see Fig. 1.3.

The level set function can be chosen to be the signed distance function from the interface. That means the smallest distance between a given point in the domain and the interface

\[
|\phi(\mathbf{r})| = d(\mathbf{r}) = \min_{\mathbf{r}_l \in I} |\mathbf{r} - \mathbf{r}_l|. \tag{1.15}
\]

The level set function is positive on one side of the interface, \( \phi(\mathbf{r}) > 0 \), and negative on the other, \( \phi(\mathbf{r}) < 0 \). To represent the density and the viscosity in the two phases one has to use a Heaviside function

\[
H(\phi) = \begin{cases} 
0, & \phi < 0, \\
1, & \phi > 0.
\end{cases} \tag{1.16}
\]

In numerical simulations the abrupt jump in the fields due to Eq. (1.16) will cause instabilities in the Finite Element Method. Therefore a smeared out Heaviside function is used instead e.g.,

\[
H_{\text{sm}}(\phi) = \begin{cases} 
0, & \phi < \epsilon, \\
\frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} \sin\left(\frac{\pi \phi}{\epsilon}\right), & -\epsilon \leq \phi \leq \epsilon, \\
1, & \phi > \epsilon,
\end{cases} \tag{1.17}
\]

Note that this is a general introduction to the level set method. The actual level set method used in this report follows the \( \frac{1}{2} \)-level set.
CHAPTER 1. THEORY

Figure 1.3: Graphical representation of the level set function. Imagine a forest fire where the gray areas represents trees and the edge is the burning fire front. Time then propagates as one moves to the right in the upper part of the figure and the fire front eats into the trees. This two phase problem (trees/burnt trees) is modeled with a level set function shown below in the figure. Observe that the propagation of the fire front is given as a level set of the level set function.

where $\epsilon$ corresponds to half the thickness of the interface. The interface thickness shall depend on the grid size in the mesh, such that it is sufficiently resolved. One can now define a new level set function

$$\tilde{\phi}(\mathbf{r}) = H_{sm}(\phi(\mathbf{r})).$$

(1.18)

The level set function in Eq. (1.18) has the advantage that it is straightforward to represent the density and viscosity in the two different phases. One simply defines scalar fields that use the level set function to distinguish between the phases

$$\rho(\mathbf{r}) = \rho_1 + (\rho_2 - \rho_1)\tilde{\phi}(\mathbf{r}),$$

(1.19)

$$\eta(\mathbf{r}) = \eta_1 + (\eta_2 - \eta_1)\tilde{\phi}(\mathbf{r}),$$

(1.20)

where the $\rho_i$ and $\eta_i$ is the density and viscosity of phase $i = 1, 2$. The interface is in this case given as the $\frac{1}{2}$-level set of Eq. (1.18). For convenience the tilde is dropped and throughout the rest of this report the level set function is interpreted as given in Eq. (1.18).

When operating with interfaces it is necessary to work with the interface normal vector

---

3Note again that the Heaviside function in Eq. (1.17) is different from the one used in this report. It just serves as an example.
and interface curvature. These two quantities are obtained from the level set function gradient and the divergence of the gradient

\[ \mathbf{n}_l = \frac{\nabla \phi}{|\nabla \phi|}, \]  \hspace{1cm} (1.21a)

\[ \kappa = -\nabla \cdot \mathbf{n}_l = -\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}, \]  \hspace{1cm} (1.21b)

where \( \mathbf{n}_l \) is the normal vector of the level set function and \( \kappa \) is the curvature. Notice that these two quantities are defined in all of the domain, \( \Omega \), and not just on the interface and domain boundary, \( \partial \Omega \). In order to maintain the thickness of the interface an intermediate step has to be performed in the numerical simulation. This step adds an artificial compression and are implemented by solving the conservation law

\[ \partial_\tau \phi + \nabla \cdot f(\phi) = 0, \]  \hspace{1cm} (1.22)

where \( \tau \) is an artificial time and \( f \) is the artificial flux. As Eq. (1.22) shall work in the regions where \( 0 < \phi < 1 \) and in the normal direction of the interface, the artificial flux is chosen to be

\[ f = \phi(1 - \phi)\mathbf{n}_l = \phi(1 - \phi)\frac{\nabla \phi}{|\nabla \phi|}. \]  \hspace{1cm} (1.23)

To avoid discontinuities at the interface, a small amount of viscosity/diffusion is added to Eq. (1.22)

\[ \partial_\tau \phi + \nabla \cdot f(\phi) = \epsilon \nabla^2 \phi, \]  \hspace{1cm} (1.24)

where \( \epsilon \) is the viscosity. When implementing Eq. (1.13) and Eq. (1.24) in COMSOL it is possible to combine the equations to one - setting the artificial time equal to the real time - such that the resulting equation for the level set function becomes

\[ \partial_t \phi + \nabla \cdot (\mathbf{v}\phi + \phi(1 - \phi)\frac{\nabla \phi}{|\nabla \phi|} - \epsilon \nabla \phi) = 0. \]  \hspace{1cm} (1.25)

where the time, \( t \), is the real time and the divergence of the artificial flux has been moved to the right-hand-side. Notice that this equation resembles a continuity equation for \( \phi \). The term with the Laplacian in Eq. (1.25) can be thought of as a kind of diffusion term, trying to enlarge the width of the interface. To counteract this diffusion one has the term with the divergence of the flux. These two terms are in equilibrium when the interface has a thickness of \( \epsilon \). Keeping the thickness of the interface constant is achieved. Eq. (1.25) can be written in conservative form

\[ \partial_t \phi + \nabla \cdot \left( \mathbf{v}\phi + \phi(1 - \phi)\frac{\nabla \phi}{|\nabla \phi|} - \epsilon \nabla \phi \right) = 0. \]  \hspace{1cm} (1.26)

Eq. (1.26) is on the form that COMSOL uses for the numerical implementation. The equation is coupled to the Navier-Stokes equation through the level set function and the velocity field. The Navier-Stokes equation reads

\[ \rho \partial_t \mathbf{v} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} = \nabla \cdot \mathbf{\sigma} + \gamma \kappa \delta(\mathbf{r} - \mathbf{r}_l)\mathbf{n}_l, \]  \hspace{1cm} (1.27)
where $\gamma$ is the surface tension. The stress tensor, $\sigma$, is for an incompressible fluid given by

$$\sigma_{ik} = -p\delta_{ik} + \eta(\partial_k v_i + \partial_i v_k),$$

(1.28)

where index notation has been used. The Kronecker delta, $\delta_{ik}$, is given by

$$\delta_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k. \end{cases}$$

(1.29)

Finally, it is possible to express the Dirac delta function as the absolute value of the gradient of the level set function

$$\delta(r - r_l) \approx |\nabla \phi|, \quad r_l \in I(t).$$

(1.30)

That Eq. (1.30) is a good approximation is intuitively clear when looking at Eq. (1.17). The absolute value of the gradient is zero for points away from the interface since the level set function is constant. As one reaches the interface the absolute value of the gradient increases, has maximum exactly at the $\frac{1}{2}$-contour of the level set function and falls off again, see Fig. 1.4. This resembles a Dirac delta function. This concludes the introduction to the level set function. This chapter is based on [5] and [6].
Chapter 2

Numerical implementation

This chapter describes the numerical implementation of the physical equations into COMSOL. First the model is set up with the necessary approximations. Then the equations are rewritten in order to implement them in the correct form. In numerics it is possible to formulate the equations and boundary conditions in strong or weak form. The strong form is what one would think of as the normal way to write the equations. The weak form is a special integral formulation of the equations that in some cases are more suited for numerical simulations. It provides the user with more possibilities to control and enforce given constraints. Nevertheless, the weak form is not as accurate as the strong form. Both forms are used to solve the present problem.

2.1 Setting up the model

The micro cuvettes can in a first approximation be thought of as two parallel plates. This geometry corresponds to the one that is analyzed in Section 1.2. Due to symmetry one only has to calculate the solution in half of the space between the plates. The full solution is obtained by mirroring in the symmetry plane. The symmetry plane is parallel with the two plates and lies half way between them. The meniscus of the fluid is assumed to be translational invariant in one direction. This direction is chosen to be the \( z \)-direction. Instead of three independent variables the problem is reduced to two independent variables, namely the \( x \) and \( y \) coordinates. In figure Fig. 2.1 one can see the domain when it is subdued to the above assumptions. The problem is reduced to two dimensions which means that COMSOL solves for four fields - the level set function, the two components of the velocity and the pressure. These fields can be written as

\[
\phi = \phi(x, y, t) \tag{2.1}
\]
\[
 v_x = v_x(x, y, t) \tag{2.2}
\]
\[
 v_y = v_y(x, y, t) \tag{2.3}
\]
\[
 p = p(x, y, t) \tag{2.4}
\]
where $p$ is the pressure. Notice that an Eulerian reference frame is used. The velocity vector field is

$$v(x, y, t) = (v_x(x, y, t), v_y(x, y, t)).$$

(2.5)

Each of the four fields has a corresponding governing equation. The level set function is governed by Eq. (1.26) corresponding to the numerical extension of the convection equation. The Navier-Stokes equation governs the two components of the velocity field. Finally, the continuity equation for mass conservation governs the pressure field. For incompressible fluids, the continuity equation reads

$$\partial_t v_i = 0.$$  

(2.6)

This gives a set of four coupled linear/non-linear differential equations. COMSOL uses a special form when solving differential equations [2]

$$d_u_i \partial_t u_i + \nabla \cdot \mathbf{G}_{u_i} = F_{u_i}, \quad \text{in } \Omega$$

(2.7)

$$- \mathbf{n}_b \cdot \mathbf{G}_{u_i} = G_{u_i} - (\partial_{u_i} R_{u_i}) \mu_{u_i}, \quad \text{on } \partial \Omega$$

(2.8)

$$0 = R_{u_i}, \quad \text{on } \partial \Omega.$$  

(2.9)
2.1. SETTING UP THE MODEL

The solution vector $\mathbf{u}$ contains

$$\mathbf{u} = \begin{bmatrix} v_x \\ v_y \\ p \\ \phi \end{bmatrix},$$

and $\mathbf{n}_b$ is the normal vector on the domain boundary. The $\mu_{ui}$’s are Lagrange multipliers. These Lagrange multipliers are only relevant when the $R_{ui}$’s are different from zero i.e. $R_{ui} \neq 0$. More generally speaking one can create a Lagrange multiplier for each constraint the problem is subjected to. The $R_{ui}$ could e.g. be equal to one of the velocity components $R_{ui} = v_x$ on a specific boundary segment. This means that the $x$-component of the velocity must be zero on that domain boundary part. Looking at Eq. (2.8) one observes that the derivative with respect to $x$-component of the velocity is different from zero. It is unity. COMSOL has the freedom to choose the Lagrange multiplier - on this special part of the boundary - such that Eq. (2.8) is fulfilled for any $G_{ui}$. The conclusion is that if a Lagrange multiplier is introduced for a given constraint in the problem then the $G_{ui}$ is irrelevant and can be left out of the treatment. One has thereby eliminated a constraint in the expense of another equation for the Lagrange multiplier.

2.1.1 Domain variables

Strong form

In the Navier-Stokes equation, Eq. (1.27), one can with advantage use the weak formulation to include the force density from the surface tension. Rewriting the remaining terms yields

$$\rho \partial_t v_i - \partial_i \sigma_{ki} = -\rho (v_i \partial_i) v_i \Rightarrow$$

$$\begin{cases} 
  d_{v_x} = \rho \\
  \mathbf{\Gamma}_{v_x} = -[\sigma_{xx}, \sigma_{xy}]^\top \\
  F_{v_x} = -\rho (v_i \partial_i) v_x \\
  d_{v_y} = \rho \\
  \mathbf{\Gamma}_{v_y} = -[\sigma_{yx}, \sigma_{yy}]^\top. \\
  F_{v_y} = -\rho (v_i \partial_i) v_y
\end{cases}$$

(2.12)

(2.13)

For the pressure one has the liberty to choose between two formulations. One where $\Gamma$ is non-zero and one where $F$ is non-zero. In this context $F$ is chosen to be non-zero

$$0 = \partial_i v_i \Rightarrow$$

$$\begin{cases} 
  d_p = 0 \\
  \mathbf{\Gamma}_p = 0 \\
  F_{p} = \partial_i v_i
\end{cases}.$$
For the level set function Eq. (1.26) is on the correct form and it is straightforward to identify the different variables

$$\partial_t \phi + \partial_i \left( v_i \phi + \phi (1 - \phi) \frac{\partial_i \phi}{|\nabla \phi|} - \epsilon \partial_i \phi \right) = 0 \Rightarrow \tag{2.16}$$

$$\begin{cases} 
    d \phi = 1 \\
    \Gamma \phi = v \phi + \phi (1 - \phi) \frac{\nabla \phi}{|\nabla \phi|} - \epsilon \nabla \phi \\
    F \phi = 0 
\end{cases} \tag{2.17}$$

This accounts for all the strong contributions in the numerical implementation concerning the domain, $\Omega$. The variables are given in the fem.equ.da, fem.equ.ga and fem.equ.f COMSOL structs and summarized in Table 2.1.

<table>
<thead>
<tr>
<th>$u_i$</th>
<th>$v_x$</th>
<th>$v_y$</th>
<th>$p$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{u_i}$</td>
<td>$\rho$</td>
<td>$\rho$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\Gamma_{u_i}$</td>
<td>$-\sigma_{xx}, \sigma_{xy}$</td>
<td>$-\sigma_{yx}, \sigma_{yy}$</td>
<td>0</td>
<td>$v \phi + \phi (1 - \phi) \frac{\nabla \phi}{</td>
</tr>
<tr>
<td>$F_{u_i}$</td>
<td>$-\rho(v_i \partial_j) v_x$</td>
<td>$-\rho(v_i \partial_j) v_y$</td>
<td>$v_i \partial_i$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.1: The variables for the strong form in the domain implemented in COMSOL.

**Weak form**

For the domain the only weak contribution comes from the Navier-Stokes equation. Therefore COMSOL only needs two equations for the velocity components. The weak contribution stems from the term including the surface tension reads

$$\gamma \kappa \delta (r - r_l) n_l. \tag{2.18}$$

One can multiply Eq. (2.18) with a test/basis function and integrate over the domain. Using Green’s theorem this integral can be divided up into a integral over the domain and one over the boundary. It is the integral over the domain that defines the weak contribution. Without going into the details the weak contributions are simply stated below

$$\gamma \left( (\partial_x \tilde{v}_x) n_{l,x} + (\partial_y \tilde{v}_y) n_{l,y} \right) |\nabla \phi| n_{l,x}, \text{ for } x, \tag{2.19}$$

$$\gamma \left( (\partial_x \tilde{v}_y) n_{l,x} + (\partial_y \tilde{v}_y) n_{l,y} \right) |\nabla \phi| n_{l,y}, \text{ for } y. \tag{2.20}$$

where the $\tilde{v}_i$’s are the so-called test/basis functions used by COMSOL. These equations are given in the fem.equ.weak struct.

**2.1.2 Boundary variables**

The boundary constraints are given by specifying the $R_{u_i}$ in Eq. (2.8) for each boundary segment. In the current set-up there is four boundary parts denoted $\partial \Omega_k$ for $k = 1..4$, see Fig. 2.1. One may write

$$R = R^{\partial \Omega_k}_{u_i}, \quad i = 1..4, \quad k = 1..4, \tag{2.21}$$
2.1. SETTING UP THE MODEL

which gives a total of 16 variables. One Lagrange multiplier is introduced for $\partial \Omega_2$ in order to enforce the contact angle condition on this boundary. This results in a weak equation on this boundary. The integral along the boundary from the transformation of Eq. (2.18) is also implemented as a weak equation.

**Strong form**

For the first boundary, $\partial \Omega_1$, the assumptions are free flow, a pressure of zero and a constant value of the level set function of one.

For the second boundary, $\partial \Omega_2$, the component of the velocity normal to the wall, $v_x$, has to be zero. The no-slip condition is relaxed such that the component of the velocity parallel to the wall are free. No conditions are applied to the pressure and the level set function.

Boundary three, $\partial \Omega_3$, has free flow like boundary one. Different simulations are performed - some with zero pressure and some with a back pressure, $\Delta p$. A constant value of the level set function of zero are assumed.

The last boundary, $\partial \Omega_4$, is the symmetry boundary. The condition for symmetry is that the component of the velocity normal to the boundary, $v_x$, is zero.

COMSOL uses the struct fem.bnd.r for the $R$ values. The values are summarized in Table 2.2. The strong form on the domain boundary is ultimately converted to weak form

<table>
<thead>
<tr>
<th>$R^{\Omega_k}$</th>
<th>$\partial \Omega_1$</th>
<th>$\partial \Omega_2$</th>
<th>$\partial \Omega_3$</th>
<th>$\partial \Omega_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_x$</td>
<td>0</td>
<td>$v_x$</td>
<td>0</td>
<td>$v_x$</td>
</tr>
<tr>
<td>$v_y$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$p$</td>
<td>$p$</td>
<td>0</td>
<td>$p - \Delta p$</td>
<td>0</td>
</tr>
<tr>
<td>$\phi$</td>
<td>$\phi - 1$</td>
<td>0</td>
<td>$\phi$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.2: The $R$ variables implemented in COMSOL. The back pressure, $\Delta p$, is the Young-Laplace pressure for the given cylindrical meniscus such that no capillary rise is expected.

in COMSOL. Nevertheless, the $R$ variables are given and then by the command flform converted from fem.bnd.r to fem.bnd.constr. This means that the $G$ variables at the same time are converted to weak equations. No $G$ variables are therefore given.

**Weak form**

The general Neumann condition is given in Eq. (2.8). In weak form this corresponds to

$$\int_{\partial \Omega} \tilde{u}_i G_{u_i} \, dr.$$  \hspace{1cm} (2.22)

The weak equation is then the integrand in Eq. (2.22). On boundary segment two, $\partial \Omega_2$, a contact angle of $\theta$ has to be applied. This boundary condition can be expressed by the dot product between the level set normal vector at the interface and the boundary normal vector

$$\mathbf{n}_l \cdot \mathbf{n}_b = \cos(\theta).$$  \hspace{1cm} (2.23)
If one finds the \( G_m \) that fulfills Eq. (2.8) and then convert the constraint to weak form by use of Eq. (2.22) the result is

\[
\begin{align*}
-\gamma |\nabla \phi| n_{l,x} \tilde{v}_x \cos(\theta), & \quad \text{for } x, \\
-\gamma |\nabla \phi| n_{l,y} \tilde{v}_y \cos(\theta), & \quad \text{for } y.
\end{align*}
\]

(2.24) (2.25)

For the Lagrange multiplier two weak equations appear

\[
\begin{align*}
\tilde{\mu} (\partial_y \phi - \cos(\theta)), & \quad (2.26) \\
\tilde{\phi} \mu, & \quad (2.27)
\end{align*}
\]

where \( \tilde{\mu} \) and \( \tilde{\phi} \) are test/basis functions for the Lagrange multiplier and the level set function. The weak equations are given in fem.bnd.weak.
Chapter 3

Results

The simulations carried out considers the capillary pump and uses the COMSOL script shown in the appendix. To verify that the script is able to keep a certain contact angle at the boundary, the interface is started as a cosine, which is unphysical due to the equations of the system. See Fig. 3.1. Besides the unperturbed interface a back pressure - corresponding to the Young-Laplace pressure - is added in order to avoid propagation of the interface. As time goes the unperturbed interface relaxes through some small oscillations. It obtains the correct shape with respect to the contact angle. In figure 3.2 is shown the water-air interface for three different times. The ends in top and bottom are open and have a zero pressure. It is therefore the pressure difference due to the surface tension, which causes the dynamics. The interface starts to be linear, but as time goes it finds the contact angle, \( \theta \), at the wall and an angle of 90° at the center of the channel, and starts to propagate. The figure considers a very short timescale and thus a Poiseuille flow has not been fully developed. The capillary propagation length can therefore not be compared to the analytical solution given in Eq. (1.8). In figure 3.3 is plotted the capillary length, \( L(t) \), as a function of time, \( t \), and for clarity also the analytical capillary length is plotted. In the time \( t \in [0; 0.5] \times 10^{-7} \) s the water-air interface accelerates to an almost linear velocity and at the time \( t = 1 \times 10^{-7} \) s the interface hits the end of the channel and the calculation is no more valid. It is expected that for further times the capillary length will follow the analytical solution. It is important to stress that the simulation is not yet verified until a simulation for longer times has been executed.
Figure 3.1: Relaxing of a cosine-perturbed water-air interface. The ends in top and bottom are closed and a back pressure corresponding to the Young-Laplace pressure is added at the top boundary. The geometry is symmetric and thus the boundary at $x = 0$ corresponds to the middle of the channel.
Figure 3.2: Capillary propagation of water-air interface. The ends in top and bottom are open and have a zero pressure. The geometry is symmetric and thus the boundary at $x = 0$ corresponds to the middle of the channel.
Figure 3.3: Capillary length, $L(t)$, as a function of time, $t$. Both the numerical and analytic results are shown, even though they are not comparable.
Chapter 4

Conclusions

A COMSOL script for simulating a two phase flow interface by using a conservative level set method has been made. It is not yet verified by comparing to analytical solutions, but it does act intuitively correct. It succeeds to keep a correct contact angle at the boundaries, which has been the most difficult part. The report does not contain very many results, but mainly describes the theory and implementation of the level set method. It has been far from trivial to implement the method, but the promising strength of the method is worth the work done.
Script linear_phi0.m

u = v_x, x-component of the velocity
v = v_y, y-component of the velocity
% Continous density
'rho', rho1+(rho2-rho1)*phi', ...

% Continous viscosity
'eta', eta1+(eta2-eta1)*phi', ...

% Start guess on the level set function
'phi0', 1/(1+exp(-(x-tan(theta)*(y-0.1E-6))/epsilon))

% Dynamic contact angle
'angle', acos(normx)

% Slip velocity
'vslip', 1e-9*4*(angle-theta)/theta*phi*(1-phi)

% Stress tensor element
'sigmaxx' '-p+2*eta*ux'
'sigmaxy' 'eta*(uy+vx)
'sigmayy' '-p+2*eta*vy'

% Shape functions
fem.shape = 'shlag(2,''u''), shlag(2,''v''), shdisc(2,1,''p''), shlag(2,''phi''), shlag(2,''lm'')

% Integration order
clear equi
fem.gporder = -4,2;

% Constraint order
clear equi
fem.cporder = -2,1;

% Equation form
fem.form = 'general';

% Subdomain settings
equ.ind = [1];
equi = -'vx','vy','p','phi'';
equ.init = [0.1;1;1;1];
equ.cpower = [1;1;1;1];
equ.gpower = [1;1;1;1];
equ.f = '-rho*(u*ux+v*uy)';
equ.ua = '-rho*(u*vx+v*vy)';
equ.ga = '-sigmaxx';
equ.weak = '-gamma*(test(ux)*normx*normx+test(vy)*normx*normy)*gphi';

% Boundary settings
clear bnd
bnd.shape = [1:4] [1:5] [1:4] [1:4];
bnd.cpower = [1;1;1;1];
bnd.gpower = [1;1;1;1];
bnd.dnr = [0:0:0;0:1];

fem.equ = equ;
clear equ
CHAPTER 4. CONCLUSIONS

fem.bnd = bnd;
clear bnd
fem = flform(fem,'outform','weak');
fem.bnd.weak = -0, ...
\(-\gamma gphi*normx*test(u)\cos(\theta)\); ...
\(-\gamma gphi*normy*test(v)\cos(\theta)\); ...
0; ...
\(test(ln)*(phiTy*cos(\theta))\); ...
\(test(\phi)*lm\); ...
0,0
fem.xmesh = meshextend(fem);
init = asseminit(fem);
tfinal = ((1.9E-6-0.3E-6)/0.4E-6)^2*3*1E-3*0.4E-6/(73e-3*cos(73/180*pi));
fem.sol=femtime(fem, 'init', init, ...
solcomp', 'u', 'phi', 'p', 'v', ...
outcomp', 'u', 'phi', 'p', 'v', ...
tlist', [0:1e-9:tfinal], ...
tout', 'tlist', 'report', 'off');
clear init
save({data dir 'data'},'fem');
Bibliography


