European Consortium for Mathematics in Industry

28th ECMI Modelling Week
Final Report

19.07.2015—26.07.2015
Lisboa, Portugal

Group 4

Shape analysis of an axisymmetric drop

Author 1: Adrián Domínguez Vázquez
Universidad Carlos III de Madrid,
Av. de la Universidad, 30, 28911 Leganés, Madrid, Spain

Author 2: Ivar Persson
Faculty of Engineering, Lund University,
Box 118, SE-221 00 Lund, Sweden

Author 3: Joana Vaz Baltazar
Instituto Superior Técnico, University of Lisbon,
1049-001 Lisboa, Portugal

Author 4: Radojka Ciganović
Faculty of Natural Sciences and Mathematics, University of Novi Sad,
Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia

Author 5: Robby Rudat
Fak. Mathematik und Naturwissenschaften, Technische Universität Dresden,
01062 Dresden, Germany

Instructor: Tihomir Ivanov
Faculty of Mathematics and Informatics, Sofia University
5 James Bourchier Blvd., 1164 Sofia, Bulgaria

Abstract

The optimal parameters that better fit the Laplacian profile corresponding to a given drop shape obtained from an image have been numerically computed succesfully. Two types of drop are considered: a pendant drop, subjected to the action of gravity only, and a rotating drop subjected to both, gravitational and centrifugal forces. An image processing algorithm is implemented to obtain the drop profile from a given image. Two numerical methods are studied: a first order Explicit Euler method and a more accurate fourth order Runge-Kutta method. According to the results, the Runge-Kutta method needs a better initial guess, being the rates of convergence of both methods identical. Ignoring the computational time, a proposed procedure would be to use the Explicit Euler method to improve a potentially bad initial guess and then use the Runge-Kutta method to pinpoint the parameters closer to their real values.

4.1 Introduction

The surface tension is a phenomenon caused by the cohesive forces among liquid molecules. In the bulk of the liquid, each molecule is pulled equally in every direction by neighboring liquid molecules, resulting in a net force of zero. But at the surface the molecules do not have the same molecules on all sides of them and therefore are pulled inwards. This creates some internal pressure and forces liquid surfaces to contract to the minimal area.

Surface tension is this elastic tendency of liquids which makes them acquire the least surface area possible. It is an important factor in the phenomenon of capillarity. Capillary is the ability of a liquid to flow in narrow spaces without the assistance of, and in opposition to, external forces like gravity. It occurs because of intermolecular forces between the liquid and surrounding solid surfaces: the combination of surface tension and adhesive forces between the liquid and container act to lift the liquid.

Numerous methodologies have been developed for the measurement of surface tension. Of these, axisymmetric drop shape analysis (ADSA) methods are considered to be the most powerful because of their accuracy, simplicity, and versatility. They are also very suitable for automated computer implementation by means of digital image analysis.

So, the goal of this project is to determine the surface tension of a drop of a certain liquid when the shape of that drop is given.

First, we will present the mathematical model relating the surface tension of a drop, subjected to the action of gravity and to rotation, with its shape. This model will be numerically solved using two different methods (Explicit Euler and Runge-Kutta). After that, when the drop's shape is given, two optimization algorithms (Newton-Gauss and Levenberg-Marquardt) will be implemented in order to obtain the optimal value of certain parameters that can be used to calculate the drop's surface tension. At the end we will use a photo of a drop and digitalize it to obtain a set of points that describes the experimental profile we have. The results will be presented and discussed.

4.2 Modelling the drop

In this section, we will follow an article by Rotenberg, Boruvka and Neumann [6] in order to present the derivation of the model describing the non rotating drop.

ADSA methods are based on the numerical fit between the experimentally obtained shapes of drops and the mathematical model given by the classical Young-Laplace equation of capillarity. This equation describes the mechanical equilibrium conditions for two homogeneous fluids separated by an interface [5]:

$$\Delta p = 2H\sigma, \tag{4.1}$$

where H is the mean curvature of the interface, σ is the surface tension and Δp is the pressure difference across the interface.

The interface is described completely as a function of the coordinates x, y and z. The graphical representation of the problem (the description of the meridional section alone) can be seen in Fig. (4.1), which is a reduced description of the system, due to its symmetry.

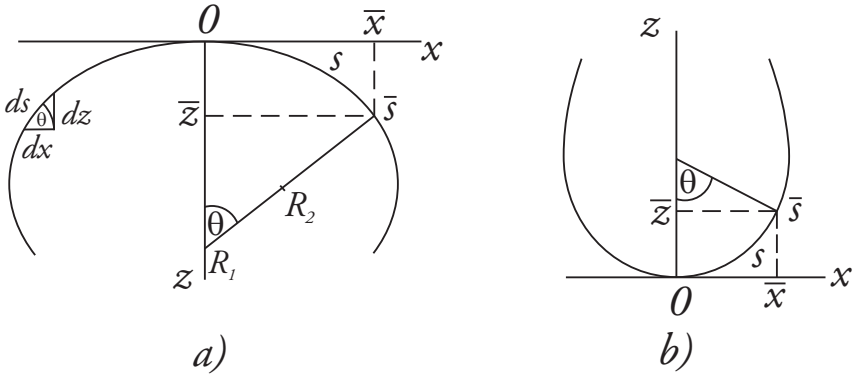

Figure 4.1: Definition of the coordinate system: a) Profile of a sessile drop; b) Profile of a pendant drop.

The Young-Laplace equation of capillarity can be written in the following way, using the fact that the curvature of a surface can be obtained by the inverse of the radius of the curvature, as shown in [6],

$$\Delta p = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right), \tag{4.2}$$

where R_1 and R_2 are the two principle radii of curvature, see Fig. (4.1).

In the absence of external forces, other than gravity, the pressure difference is due to the effect of the hydrostatic pressures and is a linear function of the elevation z, measured from the datum plane:

$$\Delta p = \Delta p_0 + \Delta\rho\, g\, z, \tag{4.3}$$

where Δp_0 the pressure difference at a selected datum plane, $\Delta\rho$ is the difference in the densities of the two bulk phases and g is the gravitational acceleration.

According to Fig. (4.1) the x axis is tangent to the curved interface and normal to the axis of symmetry and the origin is placed at the apex. Thus, we will have the following identities:

$$\begin{aligned} \Delta p_0 &= 2H_0\sigma = 2\sigma\frac{1}{R_0}, \\ \frac{1}{R_1} &= \frac{d\theta}{ds}, \\ \frac{1}{R_2} &= \frac{\sin\theta}{x}, \end{aligned} \tag{4.4}$$

where R_0 is the radius of curvature at the origin $(0,0)$, s is the arc length measured from the origin and θ is the turning angle measured between the tangent to the interface at the point (x,z) and the datum plane.

Since R_1 turns in the plane of the paper the identity for R_1 representing the rate of change of the turning angle θ with respect to the arc-length parameter s, is true by definition. On the other hand, R_2 rotates in a plane perpendicular to the plane of the paper and about the axis of symmetry.

Substituting the first of these three equations in (4.3), and then using the other two remaining equations in (4.2) we obtain:

$$\Delta p = 2\sigma\frac{1}{R_0} + \Delta\rho\, gz = \sigma\left(\frac{d\theta}{ds} + \frac{\sin\theta}{x}\right). \tag{4.5}$$

This equation can be rearranged to:

$$\frac{d\theta}{ds} = \frac{2}{R_0} + \frac{\Delta\rho\, g}{\sigma}z - \frac{\sin\theta}{x}. \tag{4.6}$$

Besides this relation, we also need to define the relations for the meridional curve. This can be represented in a parametric form

$$x = x(s), \;\; z = z(s).$$

Again using Fig. (4.1) we get the differential identities:

$$\begin{aligned} \frac{dx}{ds} &= \cos\theta, \\ \frac{dz}{ds} &= \sin\theta. \end{aligned} \tag{4.7}$$

For the three differential equations (4.6), (4.7) we consider as initial conditions

$$x(0) = z(0) = \theta(0) = 0.$$

So, the model relating the surface tension σ of a drop subjected only to gravity is given by this first-order differential system:

$$\begin{cases} \dfrac{d\theta}{ds} = \dfrac{2}{R_0} + \dfrac{\Delta\rho\, g}{\sigma} z - \dfrac{\sin\theta}{x}, \\ \dfrac{dx}{ds} = \cos\theta, \\ \dfrac{dz}{ds} = \sin\theta, \\ x(0) = z(0) = \theta(0) = 0. \end{cases} \tag{4.8}$$

Then, for given R_0 and $\Delta\rho$, σ and g the complete shape of the curve can be obtained.

If the drop is subjected to the action of gravity, as well as to rotation, with angular velocity ω, along the z axis, the problem will be similar and we still can obtain the three last equations in (4.8), but the first equation in (4.8) must have an extra term.

When the drop is rotating there exists the centrifugal force

$$\vec{f_c} = \vec{\omega} \times \vec{\omega} \times \vec{r},$$

where $\vec{\omega} = (0, 0, \omega)$ and $\vec{r} = (x(s), 0, z(s)) \equiv (x, 0, z)$.

Computing the cross vectors we get:

$$\vec{f_c} = (-\omega^2 x, 0, 0).$$

Since pressure is obtained as the quotient between a force and an area, we need to compute the following integral:

$$\int_0^x \Delta\rho\, \omega^2 u \, du = \frac{1}{2} \Delta\rho\, \omega^2 x^2.$$

Therefore, a new model is obtained:

$$\begin{cases} \dfrac{d\theta}{ds} = \dfrac{2}{R_0} + \dfrac{\frac{1}{2}\Delta\rho\, \omega^2}{\sigma} x^2 + \dfrac{\Delta\rho\, g}{\sigma} z - \dfrac{\sin\theta}{x} \\ \dfrac{dx}{ds} = \cos\theta \\ \dfrac{dz}{ds} = \sin\theta \\ x(0) = z(0) = \theta(0) = 0 \end{cases} \tag{4.9}$$

The first equation in (4.9) has several parameters, namely R_0, $\Delta\rho$, ω, σ and g. If we nondimensionalize the system we will reduce the number of parameters, simplifying the resolution of such equation, and we will avoid

the occurrence of significant errors due to the fact that the values of these parameters are of very different orders.

The second term in the equation has dimension m^{-1}, so we can nondimensionalize it by multiplying it with an arbitrary length R ($[R] \equiv \mathrm{m}$).

We define the following new variables:

$$s' = \frac{s}{R}, \;\; x' = \frac{x}{R}, \;\; z' = \frac{z}{R}.$$

From here we obtain:

$$ds = Rds', \;\; dx = Rdx', \;\; dz = Rdz'.$$

We have to do the following calculations:

- $\dfrac{d\theta}{ds} = \dfrac{d\theta}{Rds'}$ and $\dfrac{d\theta}{ds} = \dfrac{2}{R_0} + \dfrac{\frac{1}{2}\Delta\rho\,\omega^2}{\sigma}(Rx')^2 + \dfrac{\Delta\rho\, g}{\sigma}(Rz') - \dfrac{\sin\theta}{(Rx')}$

$$\Rightarrow \frac{d\theta}{ds'} = 2\frac{1}{R_0}R + \frac{\frac{1}{2}\Delta\rho\,\omega^2}{\sigma}R^3x'^2 + \frac{\Delta\rho\, g}{\sigma}R^2z' - \frac{\sin\theta}{x'}$$

- $\dfrac{dx}{ds} = \dfrac{Rdx'}{Rds'} = \dfrac{dx'}{ds'} = \cos\theta$

- $\dfrac{dz}{ds} = \dfrac{Rdz'}{Rds'} = \dfrac{dz'}{ds'} = \sin\theta$

Now, let us introduce the following notation:

$$b = \frac{1}{R_0}, \;\; d = bR, \;\; \Omega = \frac{\frac{1}{2}\Delta\rho\,\omega^2R^3}{\sigma} \;\text{ and } B = \frac{\Delta\rho\, gR^2}{\sigma}.$$

Using these identities and writing the equations denoting the variables for notational simplicity with x, z and s, instead of x', z' and s' we can rewrite the model (4.9):

$$\begin{cases} \dfrac{d\theta}{ds} = 2d + \Omega x^2 + Bz - \dfrac{\sin\theta}{x} \\ \dfrac{dx}{ds} = \cos\theta \\ \dfrac{dz}{ds} = \sin\theta \\ x(0) = z(0) = \theta(0) = 0 \end{cases} \tag{4.10}$$

4.3 Numerical Methods

Solving the ODE system

An Explicit Euler method (EE) and a fourth order Runge-Kutta method (RK4), see [1] have been implemented to get the solution of the system of ordinary differential equations (4.10) corresponding to the Young-Laplace equation of capillarity for an axisymmetric drop. These two methods give acceptable results with an affordable computational time. Implicit methods have not been considered due to the larger computational time required.

Explicit Euler (EE) method

The most simple numerical method for solving ordinary differential equations is the Euler method. Discretizing the ODE system (4.10) with the EE method, we obtain

$$\begin{aligned} \frac{x_{n+1}-x_n}{\Delta s} &= \cos\theta_n, \\ \frac{z_{n+1}-z_n}{\Delta s} &= \sin\theta_n, \\ \frac{\theta_{n+1}-\theta_n}{\Delta s} &= 2d + Bz_n + \Omega x_n^2 - \frac{\sin\theta_n}{x_n}. \end{aligned} \tag{4.11}$$

Fourth order Runge-Kutta (RK4) method

The next method we have used is the explicit fourth order Runge-Kutta method, which is actually a generalization of the Euler method. The equations describing the RK4 method are

$$\begin{aligned} k_1 &= f(s_n, y_n), \\ k_2 &= f\left(s_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ k_3 &= f\left(s_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right), \\ k_4 &= f\left(s_n + h, y_n + hk_3\right), \end{aligned}$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{4.12}$$

with s_n and y_n standing for the independent variable (s) and the dependent variables (θ, x, z) respectively, and where

$$f = \left(2d + \Omega x^2 + Bz - \frac{\sin\theta}{x}, \cos\theta, \sin\theta\right). \tag{4.13}$$

Practical estimation of the rate of convergence

The estimation of the rate of convergence is based on the assumption that c changes slowly when we change the step-size h.

Let us now look at these three equations:

$$\begin{aligned} u(x) &= y_h + ch^\alpha, \\ u(x) &= y_{\frac{h}{2}} + c\left(\frac{h}{2}\right)^\alpha, \\ u(x) &= y_{\frac{h}{4}} + c\left(\frac{h}{4}\right)^\alpha, \end{aligned}$$

where $y_h(x)$ is the numerical solution obtained with a given method using a step h, and $y_{h/2}(x)$ and $y_{h/4}(x)$ are the numerical solution given for the same method for $h/2$ and $h/4$.

If we subtract second equation from the first, and also third one from the second, we will obtain

$$\begin{aligned} 0 &= y_h + ch^\alpha - y_{\frac{h}{2}} + c\left(\frac{h}{2}\right)^\alpha \\ 0 &= y_{\frac{h}{2}} + c(\frac{h}{2})^\alpha - y_{\frac{h}{4}} + c\left(\frac{h}{4}\right)^\alpha. \end{aligned}$$

Now we have

$$\frac{y_{h/2}(x) - y_h(x)}{y_{h/4}(x) - y_{h/2}(x)} = 2^\alpha,$$

and from there we obtain the convergence rate α

$$\alpha = \log_2\left(\left|\frac{y_{h/2}(x) - y_h(x)}{y_{h/4}(x) - y_{h/2}(x)}\right|\right). \tag{4.14}$$

Optimization methods

In order to find the set of parameters corresponding to the Laplacian profile that better fits the given experimental shape of the drop, an optimization problem has to be solved. First, we will define the objective function which gives the error between the theoretical and the experimental profiles.

Definition of the error function

Suppose that $(X_m, Z_m)_{m=1}^{m=M}$ is a set of experimentally points that describe the meridional section of an interface and $(x_k, z_k)_{k=1}^{k=N}$ is a set of points from a calculated Laplacian curve. For each point of the experimental profile of the drop we search for the nearest point from the calculated Laplacian curve. The objective function to minimize is then defined as

$$E(d,\Omega,B) := \frac{1}{2}\sum_{m=1}^{M}\left[(x_n(d,\Omega,B)-X_m)^2+(z_n(d,\Omega,B)-Z_m)^2\right], \quad (4.15)$$

where (x_n, z_n), $\forall\, n \in \{1, ..., N\}$ is the nearest point from the Laplacian curve to each experimental point. The optimum set of parameters $(d_{opt}, \Omega_{opt}, B_{opt})$ must then verify

$$\begin{aligned}\frac{\partial E}{\partial d} &= 0,\\ \frac{\partial E}{\partial \Omega} &= 0,\\ \frac{\partial E}{\partial B} &= 0.\end{aligned} \quad (4.16)$$

Newton-Gauss algorithm

The first algorithm considered to solve the optimization problem is the Newton-Gauss (NG) algorithm, see [2]. This method yields the following system of linear equations to be solved iteratively,

$$\begin{pmatrix} \frac{\partial^2 E}{\partial d^2} & \frac{\partial^2 E}{\partial d\partial\Omega} & \frac{\partial^2 E}{\partial d\partial B} \\ \frac{\partial^2 E}{\partial d\partial\Omega} & \frac{\partial^2 E}{\partial \Omega^2} & \frac{\partial^2 E}{\partial \Omega\partial B} \\ \frac{\partial^2 E}{\partial d\partial B} & \frac{\partial^2 E}{\partial \Omega\partial B} & \frac{\partial^2 E}{\partial B^2} \end{pmatrix}_{(d_k,\Omega_k,B_k)} \begin{pmatrix} \Delta d \\ \Delta\Omega \\ \Delta B \end{pmatrix}_k = - \begin{pmatrix} \frac{\partial E}{\partial d} \\ \frac{\partial E}{\partial \Omega} \\ \frac{\partial E}{\partial B} \end{pmatrix} \quad (4.17)$$

with,

$$\begin{pmatrix} \Delta d \\ \Delta\Omega \\ \Delta B \end{pmatrix}_k = \begin{pmatrix} d_{k+1}-d_k \\ \Omega_{k+1}-\Omega_k \\ B_{k+1}-B_k \end{pmatrix} \quad (4.18)$$

The iterative process starts with an initial guess (d_0, Ω_0, B_0) and stops when the correction $(\Delta d, \Delta\Omega, \Delta B)^k$ is below a given tolerance.

The second derivatives of the error function are computed numerically using second order divided differences, given by the following general expression for a function $g : \mathbb{R} \to \mathbb{R}$, a certain variable x and a step h:

$$\frac{\partial^2 g}{\partial x^2} = \frac{g(x+h)-2g(x)+g(x-h)}{h^2} \quad (4.19)$$

Thus, for instance, the second derivative of the error function for the variable d is given by

$$\frac{\partial^2 E}{\partial d^2} = \frac{E(d+h,\Omega,B) - 2E(d,\Omega,B) + E(d-h,\Omega,B)}{h^2} \tag{4.20}$$

For B and Ω will be similar.

The cross derivatives are approximated, for a function $g : \mathbb{R} \to \mathbb{R}$, a certain variable x and a step h, using the formula

$$\frac{\partial^2 g}{\partial x \partial y} = \frac{g(x+h_x, y+h_y) - g(x+h_x, y) - g(x, y+h_y) + g(x,y)}{h_x, h_y} \tag{4.21}$$

where x and y stands for general independent variables and h, h_x and h_y the steps considered for each of the variables, respectively.

Thus, for example, for Ω constant $\frac{\partial^2 E}{\partial d \partial B}$ will be equal to

$$\frac{\partial^2 E}{\partial d \partial B} = \frac{E(d+h_d,\Omega,B+h_b) - E(d+h_d,\Omega,B) - E(d,\Omega,B+h_B) + E(d,\Omega,B)}{h_d, h_B} \tag{4.22}$$

The other cross derivatives will be similar.

Levenberg-Marquardt algorithm

One of the most used optimization algorithm is Levenberg-Marquardt (LM). Our aim is to minimize error function, which is given with (4.15). If we put

$$err(x_n(d,\Omega,B), z_n(d,\Omega,B)) = \left[(x_n(d,\Omega,B) - X_m)^2 + (z_n(d,\Omega,B) - Z_m)^2\right], \tag{4.23}$$

then we resolve our problem with the folowing sheme:

$$(d_{n+1}, \Omega_{n+1}, B_{n+1}) = (d_n, \Omega_n, B_n) + (J^T J + \lambda * I)^{-1} J^T err(xCalc, zCalc), \tag{4.24}$$

where $\lambda > 0$, I is identity matrix and J is Jacobian from r, that is

$$J = \Delta r = \begin{pmatrix} \frac{\partial r}{\partial d} & \frac{\partial r}{\partial \omega} & \frac{\partial r}{\partial B} \end{pmatrix} \tag{4.25}$$

Obtaining the initial guess

The iterative methods presented above for solving the optimization problem need an initial guess (d_0, Ω_0, B_0) in order to start the optimization process. This guess must be computed from the given experimental data. In order to do that, we find a polynomial that fits a set of points $\{x_1, ...x_L\}$ from the experimental profile of the drop including its apex, which corresponds to the origin of the coordinate system. Since the drop is axisymmetric,this polynomial only have even terms, as for example,

$$p(x) = ax^4 + bx^2. \tag{4.26}$$

Using this polynomial, we can define an objective function to optimize in order to get the initial guess by considering the right hand side of the first equation in (4.10). The objective function is then defined as,

$$\Phi(d, \Omega, B) = \sum_{l=1}^{L} \left(g(d, \Omega, B, x_l, , \theta_l, z_l) - y_l\right)^2 \tag{4.27}$$

where z_l and θ_l are computed using the fitted polynomial,

$$z_l = p(x_l), \tag{4.28}$$

$$\theta_l = \tan^{-1}\left(\frac{dz}{dx}\right) = \tan^{-1}\left(\frac{dp(x_l)}{dx}\right), \tag{4.29}$$

$g(d, \Omega, B, x_l, \theta_l, z_l)$ is the right hand side of the first equation in (4.10) evaluated in x_l, z_l and θ_l and the values y_l are defined as,

$$y_l = \frac{d\theta}{ds}_{|x_l} = \frac{d\theta}{dx}_{|x_l}\frac{dx}{ds}_{|x_l} = \frac{1}{1 + \left(\frac{dp(x_l)}{dx}\right)^2}\frac{d^2p(x_l)}{dx^2} \tag{4.30}$$

The problem to be solved is then,

$$\begin{aligned} \frac{\partial \Phi}{\partial d} &= 0, \\ \frac{\partial \Phi}{\partial \Omega} &= 0, \\ \frac{\partial \Phi}{\partial B} &= 0, \end{aligned} \tag{4.31}$$

which leads to the following linear system of equations,

$$\begin{pmatrix} \sum_{l=1}^{L} 4 & \sum_{l=1}^{L} 2x_l^2 & \sum_{l=1}^{L} 2z_l \\ \sum_{l=1}^{L} 2x_l^2 & \sum_{l=1}^{L} x_l^4 & \sum_{l=1}^{L} x_l^2 z_l \\ \sum_{l=1}^{L} 2z_l & \sum_{l=1}^{L} x_l^2 z_l & \sum_{l=1}^{L} z_l^2 \end{pmatrix}_{(d_k, \Omega_k, B_k)} \begin{pmatrix} \Delta d \\ \Delta \Omega \\ \Delta B \end{pmatrix}_k = - \begin{pmatrix} \sum_{l=1}^{L} 2\left[y_l + \frac{\sin\theta_l}{x_l}\right] \\ \sum_{l=1}^{L} x_l^2\left[y_l + \frac{\sin\theta_l}{x_l}\right] \\ \sum_{l=1}^{L} z_l\left[y_l + \frac{\sin\theta_l}{x_l}\right] \end{pmatrix} \tag{4.32}$$

These are the equations for the most general case. The equations can be changed to fit allready known parameters. For example, $\Omega = 0$ leads to

$$\begin{pmatrix} \sum_{l=1}^{L} 4 & \sum_{l=1}^{L} 2x_l^2 & \sum_{l=1}^{L} 2z_l \\ 0 & 1 & 0 \\ \sum_{l=1}^{L} 2z_l & \sum_{l=1}^{L} x_l^2 z_l & \sum_{l=1}^{L} z_l^2 \end{pmatrix}_{(d_k, \Omega_k, B_k)} \begin{pmatrix} \Delta d \\ \Delta \Omega \\ \Delta B \end{pmatrix}_k = - \begin{pmatrix} \sum_{l=1}^{L} 2\left[y_l + \frac{\sin\theta_l}{x_l}\right] \\ 0 \\ \sum_{l=1}^{L} z_l\left[y_l + \frac{\sin\theta_l}{x_l}\right] \end{pmatrix} \tag{4.33}$$

In order to get a better estimation of the initial guess, specially for pendant drops, this process is split into two steps. In the first step, the objective is to obtain d. Ω and B are set to zero and the polynomial is fitted using a set of points located around the apex. Once d is obtained, it is fixed for the second step and Ω and B are computed by fitting the polynomial to a wider part of the drop shape.

Image processing

In the sections before all calculations were based on a given curve, but in the basic problem was to do the calculations based on a given picture. To close the gap between the picture and the curve, a picture-processing algorithm was implemented. This algorithm imports a given picture like picture 4.2 and converts it into a grey scaling picture, which is saved as a matrix with the size of the resolution of the picture.

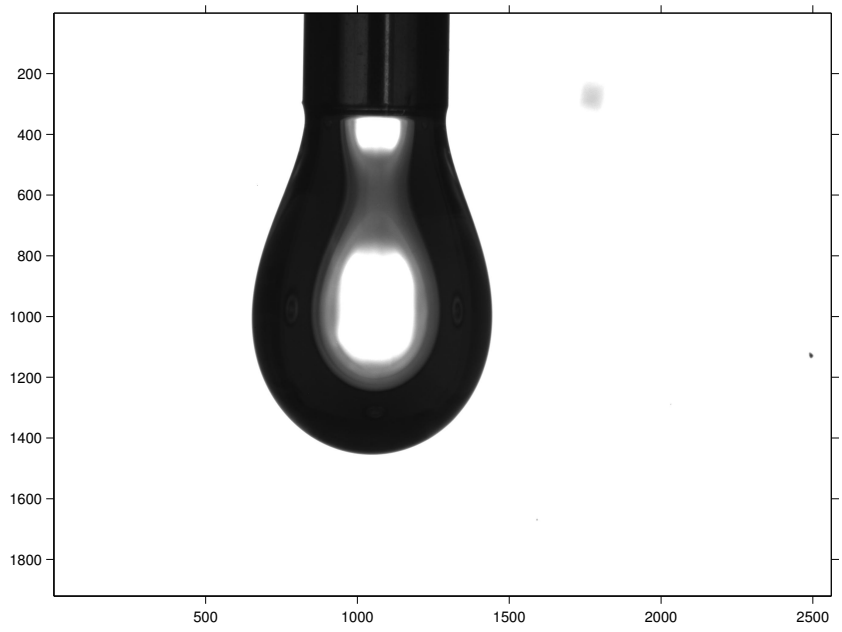

Figure 4.2: Picture of a pendant drop.

Given a threshold value, that matrix is converted to a binary matrix, dividing the values under and above the threshold. Also an a algorithm is implemented to fix the reflections on the drop, to get an black and white picture like in picture 4.3 .

In the end the algorithm is crawling around the drop, collecting the indices of the pixel on the border. To get an even shape only every j^{th} pixel is taken, where j is a number depending on the resolution of the picture.

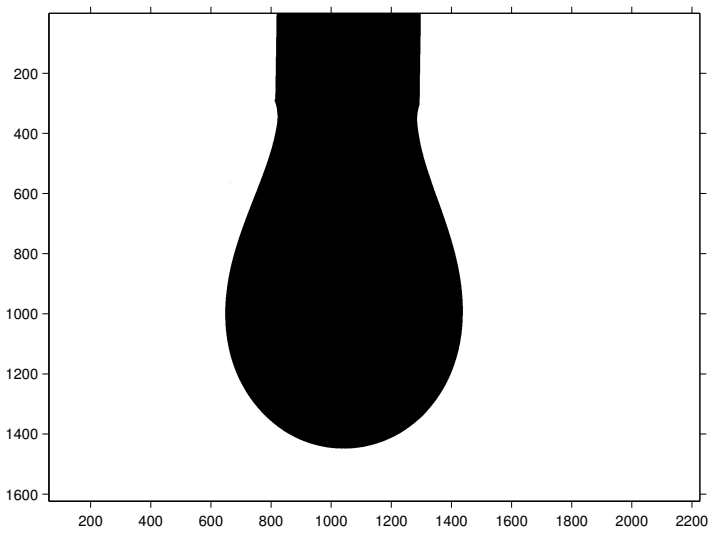

Figure 4.3: Binary black and white picture

Given the wide of the tube the indices are finally transformed into the needed coordinates.

4.4 Numerical Results

The numerical results were obtained by functions implemented in Matlab.

Solving the ODE system

EE and RK4 methods were tested for two sets of values for the model parameters:

1. Pendant drop: $\Omega = 0$, $d = 2.0644$ and $B = -2.4$;
2. Rotating drop: $\Omega = -1.873$, $d = 1.504438$ and $B = 0$.

The shape, (x, z), of the drop obtained from both methods was pratically the same, as it can be seen in Fig. (4.4).

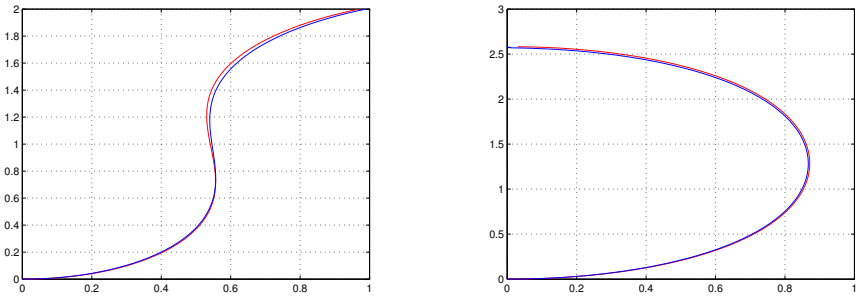

Figure 4.4: Shape obtained from EE and RK4.

The convergence rates of two methods were estimated using eq. (4.14), and are represented in Fig. (4.5) and (4.6).

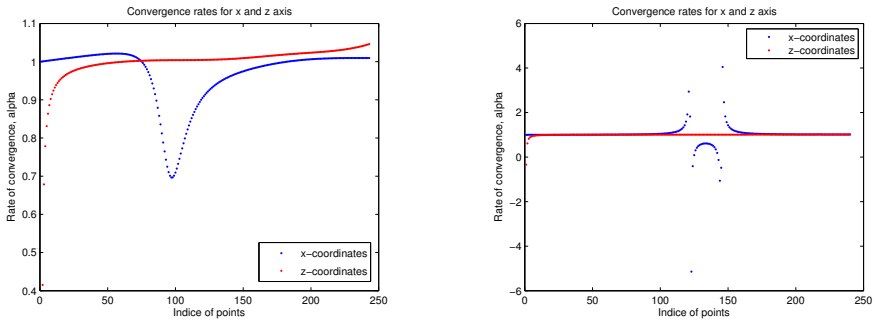

Figure 4.5: Rate of convergence of EE and RK4, respectively, for the pendant drop.

The rates of convergence are very similar, though it was expected that RK would have 4th order convergence.

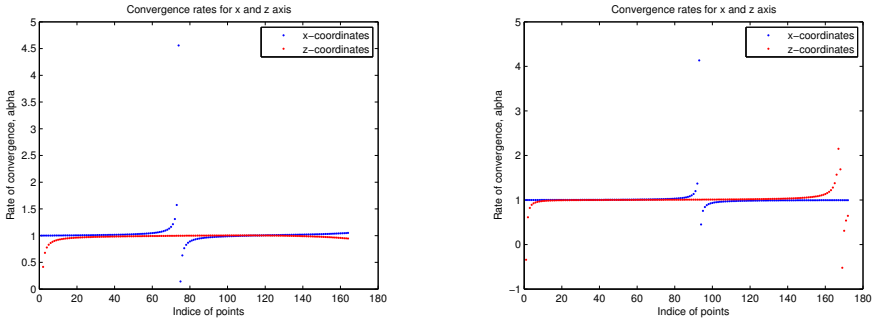

Figure 4.6: Rate of convergence of EE and RK4, respectively, for the rotating drop.

Optimization methods

We were given several sets of experimental data, which we used as input for NG and LM algorithms. Here we just present the results for the two sets corresponding to the pendant and rotating drops mentioned.

The initial values selected for the experimental data set of the pendant drop were $\Omega = 0$, $d = 2$ and $B = -2.3$, and for the rotating drop they were $\Omega = -1.85$, $d = 1.5$ and $B = 0$. These values were very close to the real ones.

For these initial values, the values obtained for Ω, d and B, which minimized the error function (4.15), are indicated in the following tables.

Pendant drop

	NG EE	NG RK	LM EE	LM RK
Ω	0	0	0	0
d	2.0584	2.0642	2.003699	2.040388
B	-2.4208	-2.3997	-2.300573	-2.308510
time (s)	0.14195	0.4637	0.9774	1.2538

Rotating drop

	NG EE	NG RK	LM EE	LM RK
Ω	-1.8932	-1.8747	-1.849561	-1.850084
d	1.5099	1.5048	1.499807	1.499031
B	0	0	0	0
time (s)	0.18803	0.82283	1.8028	3.0245

From these results we can easily conclude that the computational time for the RK method is considerably longer than the one for the EE method, with both NG and LM algorithms.

Fig. (4.7)-(4.10) represent the shape of the drop for these values.

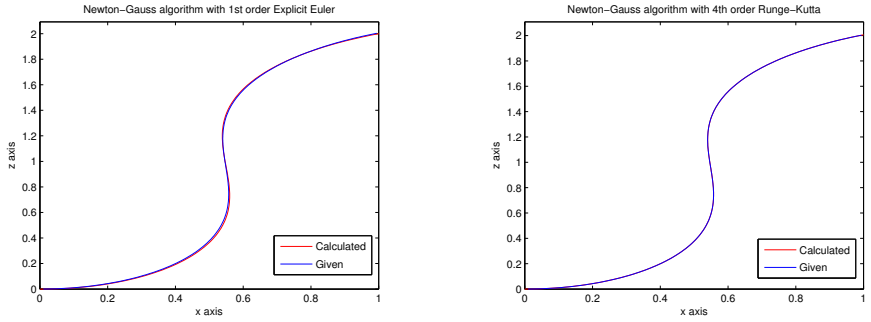

Figure 4.7: Comparing the shape obtained by NG with EE and RK4, respectively, for the pendant drop, with tolerance 0.01.

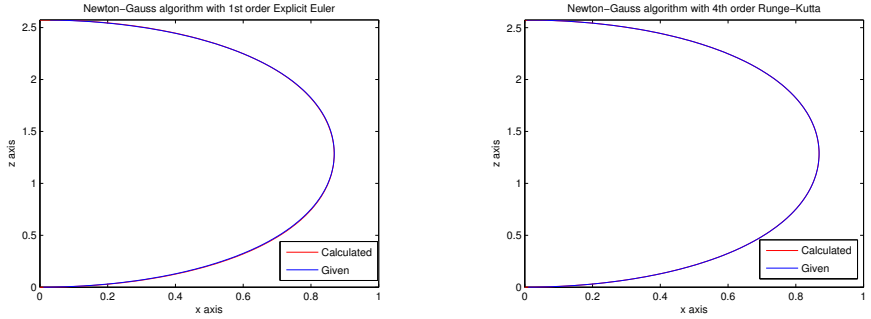

Figure 4.8: Comparing the shape obtained by NG with EE and RK4, respectively, for the rotating drop, with tolerance 0.0001.

In Fig. (4.10) for EE, the computational time for the experiment with tolerance equal to 0.0001 was considerably bigger than the time for the experiment with tolerance equal to 0.01, so only the last is presented.

Obtaining the initial guess

The implemented basic version of the initial guess algorithm gives mixed results. If the initial guess was good enough was depending on the algorithm that was using it and the part of the drop that was used to obtain the guess.

For the different algorithms and the given theoretical curves, there was no real system to be seen if it was converging with the initial guess or not, but for every curve there was at least one algorithm that was converging.

But for the part of the curve it seemed that it was depending on the type of drop that was given. For pendant drops it gave better results to use a bigger part of the drop, where for rotating drops the results with smaller parts of the curve were better.

For the curve obtained from the image processing the basic version didn't give a satisfying result. But with the two step version, it was possible to

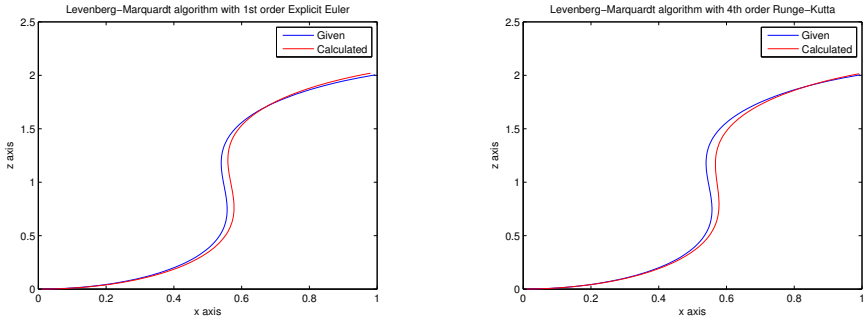

Figure 4.9: Comparing the shape obtained by LM with EE and RK4, respectively, for the pendant drop, with tolerance 0.01.

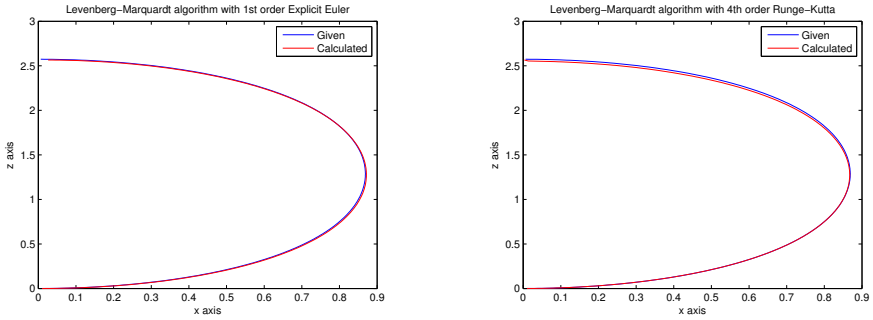

Figure 4.10: Comparing the shape obtained by LM with EE and RK4, respectively, for the rotating drop, with tolerances 0.01 and 0.0001, respectively.

obtain an initial guess, which was good enough, that the following algorithm were converging and approximating the curve very well.

Image Processing Algorithm

The image-processing-algorithm implemented during the project, successfully obtained the curve from the given picture. But for a good result a high resolution picture is needed. If the resolution is too low, the curve will be not smooth or you will obtain not enough points for a satisfying result. For example the figure 4.2 with a resolution of 2560x1920 gives a curve with 226 points and with a resolution 1280x960 it gives a curve with 282 points (see Fig. (4.11)).

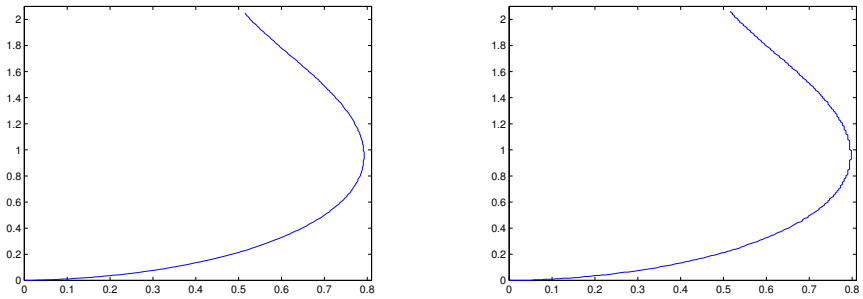

Figure 4.11: Comparing the shape obtained from high and low resolution picture, respectively.

4.5 Conclusions

The optimal parameters that better fit the Laplacian profile corresponding to a given drop shape have been computed successfully.

As we can see in the results all methods produce results that are reasonably good with initial conditions that are good enough. We have assessed the stability and the computational times of both methods. The Explicit Euler method is more stable than the Runge-Kutta method and it converges without the need of a so accurate initial guess in comparison to the Runge-Kutta method. Even though the rates of convergence for the Explicit Euler and the Runge-Kutta method are the same, the latter produces more accurate results. Ignoring the computational time, a proposed procedure would be to use the Explicit Euler to improve a potentially bad initial guess and then use the Runge-Kutta to pinpoint the parameters closer to their real values.

Bibliography

[1] Butcher, JC. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 2008, second edition.

[2] J. Nocedal, SJW. *Numerical Optimization*. Operations Research. Springer, 1999.

[3] Ranganathan, A. The Levenberg-Marguardt Algorithm June 2004.

[4] Wikipedia. Surface tension.
URL `https://en.wikipedia.org/wiki/Surface_tension`

[5] ———. Young-laplace equation.
URL `https://en.wikipedia.org/wiki/Young-Laplace_equation`

[6] Y. Rotenberg, LB and Neumann, AW. Determination of surface tension and contact angle from the shapes of axisymmetric fluid interfaces. *Journal of Colloid and Interface Science* **93**(1), 169–183, May 1983.