Vaya Sapobi Samui Vos

Deformations of a bubble

Bachelor's thesis, 2 April 2012

Thesis supervisors: Prof.dr. M. van Hecke, Dr. V. Rottschäfer

Mathematisch Instituut
Leids Instituut voor Onderzoek in Natuurkunde
Universiteit Leiden

Contents

1 Introduction

The Granular and Disordered Media group at Leiden University investigates the slow flow and jamming of granulates, suspensions and foams. They work a lot with foam bubbles and so far usually approximate the relation between the force F applied on the bubble and the deformation δ of the bubble to be a linear relation: $F = K\delta$, where K is a certain constant. Every contact between bubbles is approximated as though they are springs. For very small bubbles with only one side being pressed, we find that $K = 4\pi\sigma$. Here σ is the surface tension with the dimension of force per unit length, or of energy per unit area. It is a constant that depends on the cohesion and adhesion of the fluids which are separated by the surface. But then we wonder what we mean with a very small bubble? Apart from that, the bubble used in the experiments are not really small at all and usually have more than one side being pressed by other bubbles. The linear relation might not even hold for such case. In order to understand what is really going on with the bubbles, we have decided to look more closely at a single bubble.

In this thesis we are going to look at a single air bubble in a fluid with added surfactant pressed up against a flat plate by a buoyant force. This will give a curved blob with a circular contact-area of radius r_c against the plate (See figure 1). We will characterize the bubble by its volume and say that a bubble has size R if its volume is equal to the volume of a sphere of radius R. Our main goal is to have some theory for this case using known theories on this subject and to see whether or not these theories make sense by simulating the bubble shape. But before being able to do that we first have to assemble the theory and find out how the simulation of the bubble goes (using a simulator called Surface Evolver). And that is what the reader can expect in this thesis.

In the next chapter we will provide the reader with some background theory about fluids, the buoyant force and what determines the shape of a bubble. With the theories found, we have approximated certain aspects of the bubble such as its shape, the radius of the contact-area r_c and the linear force-deformation relationship stated above. This has been done for very small bubble $R \ll l_c$ and very large bubbles $R \gg l_c$. Here l_c is the characteristic length which is defined as $l_c = \sqrt{\frac{\sigma}{\Delta\rho g}}$. Where $\Delta\rho$ is the density difference of the substances which create the bubble, g the gravitational constant and σ the surface tension. For calculation purposes we use $\Delta\rho = 1070 kg/m^3$, $g = 9.81 N/kg$ and $\sigma = 28 \cdot 10^{-3} N/m$, (the value for σ for the surfactant DAWN) which gives $l_c \approx 1.63$mm.

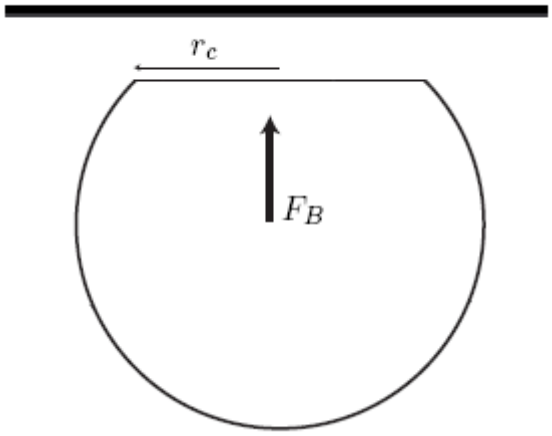

Figure 1: Front view of a single bubble pressed up against a flat plate by buoyant force F_B with contact-radius r_c.

The approximations are put together in chapter 3 of this thesis. In chapter 4 we will explain how the simulator Surface Evolver works, how we use it and we will also show some data we got from the simulations.

We end the thesis with some concluding remarks and suggestions for further research.

2 Theory

The assembled theory which we present in this chapter could be found back mainly in [5],[6], [7] and [3].

2.1 Buoyant force

In the introduction we have said that we are going to look at a bubble pressed up against a flat plate by a buoyant force, but what exactly is this buoyant force?

Archimedes' principle states that
"The force of buoyancy equals (minus) the weight of the displaced fluid"

Let us consider a body with volume V and density ρ_{body} which is completely in a fluid with density ρ_{fluid}. The densities need not to be uniform over the whole volume. There are two forces working on the body, both in the z-direction: The gravitational force F_G and the Buoyant force F_B. These are given by:

$$F_G = \int_V -\rho_{body}\ g\ dV \text{ and } F_B = \int_V \rho_{fluid}\ g\ dV.$$

Here g is the gravitational acceleration constant.
The resultant force on the body is thus

$$F = F_G + F_B = \int_V (\rho_{fluid} - \rho_{body}) g\ dV.$$

For bodies and fluids of uniform densities such as the bubble we are looking at, the formula simplifies to

$$F = \Delta\rho g V \tag{2.1}$$

where $\Delta\rho$ is the density difference between the body and the fluid.

2.2 The Young-Laplace equation

What determines the shape of a gas bubble in a liquid? If the pressures inside and outside of the bubble are constant and no external forces are applied on the bubble, then the shape of the bubble is determined solely by the surface tension σ. The bubble will take the shape that would minimize the surface area so that the energy from surface tension is minimal. This shape is known to be the spherical shape.

However, the pressures inside and outside of the bubble are not constant due to gravity. The pressure in a substance with uniform density $\rho_{substance}$ changes with depth by the following formula:

$$p(z) = p_0 + \rho_{substance}\ gz \tag{2.2}$$

Here g is the gravitational acceleration constant, p_0 the pressure at a certain reference height and z the depth measured from the reference height.

In general the shape of an air bubble in a liquid, that is not pressed up against anything, is given by the Young-Laplace equation. The equation states the relationship between the pressure drop across the interface Δp at a certain point, the surface tension σ and the curvature of the interface $\frac{1}{R_1} + \frac{1}{R_2}$ at the same point. This equation is given as follows:

$$\Delta p = p_{in} - p_{out} = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right)$$

Here p_{in} is the pressure inside the bubble and p_{out} the pressure outside the bubble. R_1 and R_2 are called the principal radii of curvature and will be explained together with the curvature in the next subsection. For now we would like to point out that the curvature of very curved surfaces is high, and the curvature of a flat surface is zero. For a spherical surfaces of radius R_0, both of the principal radii of curvature have the same value R_0, so the formula reduces to

$$\Delta p = \frac{2\sigma}{R_0}$$

The pressure drop Δp is sometimes also referred to as the capillary pressure p_c.

If we try to apply the Young-Laplace equation on our bubble pressed up against a plate, we will see that something is wrong at the contact-area of the bubble and the flat plate. Since the surface is flat there, the curvature is zero and so the Young-Laplace equation states that the pressure difference at the contact-area is zero. But that is certainly not the case, so there must be something else.

The shape of the whole bubble is still given by the Young-Laplace's law, but something tricky is going on at the surface of the contact-area. At the film between the bubble and the flat plate the pressure outside of the bubble is not simply the pressure in the liquid p_l, but a sum of the liquid pressure and a so called disjoining pressure $\prod$(h), which is a function of the film thickness h (See figure 2). The disjoining pressure results form attractive and repulsive forces in the thin film, such as electrostatic and steric repulsions. So we have $p_{out} = p_l + \prod$ at the thin film between the contact-area and the flat plate and $p_{out} = p_l$ elsewhere. Since the curvature at a flat film is zero, we see from the Young-Laplace equation that the disjoining pressure balances out the pressure difference between the gas in the bubble and the liquid outside of the bubble $p_{in} - p_l = \prod$.

At equilibrium the driving force is balanced by the disjoining pressure in the thin film on the contact-area, giving us the equation:

$$F = \pi r_c^2 \prod . \tag{2.3}$$

2.3 Curvature

To use the Young-Laplace equation in more detail, we need to know what the curvature $\frac{1}{R_1} + \frac{1}{R_2}$ is. In this subsection we will describe the curvature of a surface and include a mathematical description of it.

Consider a smooth surface with arbitrary shape. For a given point on the surface, we have an infinite amount of planes , containing the normal, intersecting the surface. Note each of such planes is just a rotation around the normal of another plane containing the normal. For each of these planes, the intersection between the plane and the surface is a smooth curve. Nearby the given point, this curve can be approximated by a part of a circle with its center on the normal of the surface. The center of this circle is called the center of curvature and its radius is called the radius of curvature of the intersection. Depending on

which side of the surface the center is located, the radius can have a positive or a negative sign. If we started out with one certain plane and rotate it around the normal, one would find that the radius of curvature varies between a maximum and minimum value of the radius of curvature R_1 and R_2. These two radii are called the principal radii of curvature. In figure 3 we see the circles that gives the principal radii of curvature for a saddle shaped surface. According to [6, p.99] it can be shown that the planes from which we obtain the two principal radii of curvature are orthogonal to each other and that the radius of curvature on any other intersection plane can be determined from the radii of curvature. So locally the shape of a surface is defined completely by the principal radii of curvature. For a surface of a sphere with radius R_0 we thus have R_1=R_2=R_0.

Rather than using the principal radii or curvature to define the shape of a surface one often uses the following scalar quantities: the Gauss curvature K and the mean curvature H defined by

$$K = \frac{1}{R_1 R_2}$$

$$H = \frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$$

Note that in certain books and websites the mean curvature is defined as $H = \left(\frac{1}{R_1} + \frac{1}{R_2}\right)$ instead. Since we are only interested in the mean curvature for the Young-Laplace equation, we will set the Gauss curvature aside from now on.

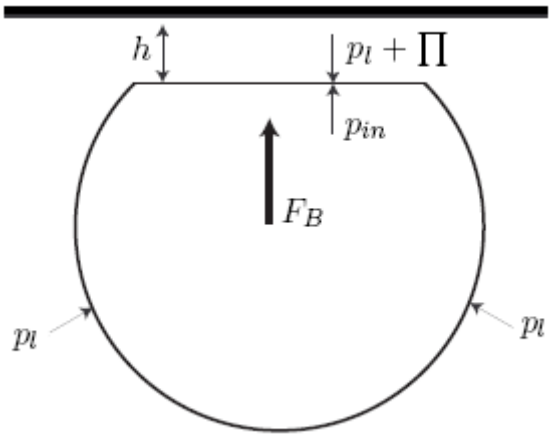

Figure 2: Front view of a bubble pressed up against a flat plate showing the pressures at each part of the bubble.

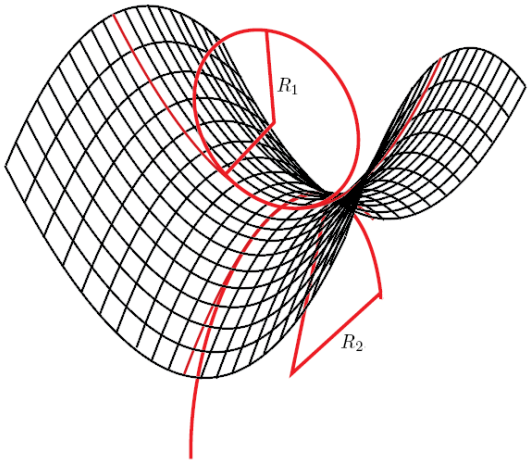

Figure 3: A saddle surface showing the idea of the principal radii of curvature R_1 and R_2. Image modified from [4, p.21]

2.3.1 Mathematical description

Mathematically for a surface in the 3-dimensional Euclidean space $\mathbb{R}^3$, the mean curvature is given by the trace of the second fundamental form of Gauss, which is a symmetric tensor. In this subsubsection we will give the definitions needed to get the second fundamental form of Gauss following the treatments of [3] and some notations of [9].

A surface S is defined as a two-dimensional (smooth) object embedded in d-dimensional Euclidean flat space $\mathbb{R}^d$. To describe the surface we can cover it by "patches" U_α such that every point of the patch may be labeled by two coordinates

$$\underline{\sigma} = (\sigma^i; i = 1, 2)$$

in some subset V of $\mathbb{R}^2$. The position in $\mathbb{R}^d$ of a point P in patch U_α with coordinate $\underline{\sigma}$ can then be represented by a d component vector

$$\vec{X}(\underline{\sigma}) = (X^\mu(\underline{\sigma}); \mu = 1, \ldots, d)$$

A basis for the plane tangent to S at point P is given by the two tangent vectors

$$\vec{t_i} = \frac{\partial \vec{X}(\underline{\sigma})}{\partial \sigma^i}, \quad i = 1, 2$$

Since we stated that the curvature would be given by the trace of a tensor let us also give a short explanation of what a tensor is.
A $\binom{P}{Q}$ tensor T is an object defined by its 2^{P+Q} components in a coordinate system $\underline{\sigma}$. The $\binom{P}{Q}$ notation shows that the tensor T has P upper indices and Q lower indices.

$$T^{i_1 \ldots i_P}_{j_1 \ldots j_Q}$$

The tensor T has the property that under a change of coordinates $\underline{\sigma} \to \underline{\sigma}'$, the components of T transform into T' according to

$$T'^{k_1 \ldots k_P}_{l_1 \ldots l_Q} = \frac{\partial \sigma'^{k_1}}{\partial \sigma^{i_1}} \cdots \frac{\partial \sigma'^{k_P}}{\partial \sigma^{i_P}} \cdot \frac{\partial \sigma^{l_1}}{\partial \sigma'^{j_1}} \cdots \frac{\partial \sigma^{l_Q}}{\partial \sigma'^{j_Q}} \cdot T^{i_1 \ldots i_P}_{j_1 \ldots j_Q}$$

The tensors we are going to use in this section are very much like a vector of a matrix known in linear algebra. The tangent vectors t_i are of tensors of type $\binom{0}{1}$. This type of tensors are sometimes also called a covariant vector.

We will now define certain tensors and operations needed later on. We keep on using the same notations for coordinates $\underline{\sigma}$, position of a point in the plane $\vec{X}$ and the tangent vectors $\vec{t_i}$ as defined above.

The metric tensor
The metric tensor or the first fundamental form is defined by

$$\begin{aligned} g_{ij}(\underline{\sigma}) &= \frac{\partial \vec{X}}{\partial \sigma^i} \frac{\partial \vec{X}}{\partial \sigma^j} \\ &= \vec{t_i} \cdot \vec{t_j} \end{aligned}$$

for $i, j = 1, 2$. Here $\vec{t_i} \cdot \vec{t_j}$ is just the vector inner product.

This is a type $\binom{0}{2}$ tensor for which its integral gives the square of the infinitesimal Euclidean distance dS between two points with respective coordinates $\underline{\sigma}$ and $\underline{\sigma} + d\underline{\sigma}$. For

$$\begin{aligned} dS^2 = (\vec{X}(\underline{\sigma} + d\underline{\sigma}) - \vec{X}(\underline{\sigma}))^2 &= d\sigma^i d\sigma^j \frac{\partial \vec{X}}{\partial \sigma^i}(\underline{\sigma}) \frac{\partial \vec{X}}{\partial \sigma^j}(\underline{\sigma}) \\ &= d\sigma^i d\sigma^j g_{ij}(\underline{\sigma}) \end{aligned}$$

If we treat the metric tensor as a 2×2 matrix and look at its inverse, we will have a type $\binom{2}{0}$ tensor g^{ij} defined by the elements of the inverse matrix.

$$g^{ij} = (g_{kl}^{-1})_{ij}.$$

Do note that both these tensors are symmetric, so $g_{ij} = g_{ji}$ and $g^{ij} = g^{ji}$ for all $i, j \in \{1, 2\}$.

Christoffel symbols
The symbols

$$\begin{aligned} [jk, l] &= \frac{1}{2}\left(\frac{\partial g_{jl}}{\partial \sigma^k} + \frac{\partial g_{lk}}{\partial \sigma^j} - \frac{\partial g_{jk}}{\partial \sigma^l}\right) \\ \Gamma^i_{jk}(\underline{\sigma}) &= \sum_l g^{il}[jk, l] \end{aligned}$$

are called the Christoffel symbols of first kind and second kind respectively. Despite the notation of the Christoffel symbol of the second kind, it is not a tensor.

Covariant derivative
For a $\binom{m}{n}$ tensor $T^{i_1 \cdots i_m}_{j_1 \cdots j_n}$, the covariant derivative of T with respect to σ^k is defined by:

$$D_k T^{i_1 \cdots i_m}_{j_1 \cdots j_n} = \frac{\partial T^{i_1 \cdots i_m}_{j_1 \cdots j_n}}{\partial \sigma^k} - \sum_{\alpha=1}^{n} \sum_s \Gamma^s_{j_\alpha k} T^{i_1 \cdots i_m}_{j_1 \cdots j_{\alpha-1} s j_{\alpha+1} \cdots j_n} + \sum_{\beta=1}^{m} \sum_s \Gamma^{i_\beta}_{ks} T^{i_1 \cdots i_{\beta-1} s i_{\beta+1} \cdots i_m}_{j_1 \cdots j_n}$$

So for a $\binom{0}{1}$ tensor such as the tangent vectors $\vec{t_i}$ we get that the covariant derivative of $\vec{t_i}$ with respect to σ^k is

$$D_k \vec{t_i} = \frac{\partial \vec{t_i}}{\partial \sigma^k} - \sum_s \Gamma^s_{ki} \vec{t_s} \tag{2.4}$$

Extrinsic curvature tensor
The extrinsic curvature tensor $\vec{K_{ij}}$ is defined by

$$\vec{K_{ij}} = D_i \vec{t_j} \tag{2.5}$$

and it is a symmetric $\binom{0}{2}$ tensor with each component normal to the surface. In the particular case of a surface in $\mathbb{R}^3$, $\vec{K_{ij}}$ is proportional to the unit normal vector $\vec{n}$ and one can write it as

$$\vec{K_{ij}} = K_{ij} \vec{n} \tag{2.6}$$

where K_{ij} is a symmetric tensor called the second fundamental form of Gauss.

The mean curvature
The mean curvature H is defined by:

$$H = \frac{1}{2} \sum_i \sum_j g^{ij} K_{ij}.$$

2.3.2 Example for surfaces with rotational symmetry

Since we are interested in the shape of an air bubble pressed up against a flat plate by buoyant force only, we have a case of a surface with rotational symmetry in $\mathbb{R}^3$. In this subsubsection we will derive the expression of curvatures for surfaces with rotational symmetry using the notations from the previous subsubsection. Let us use the polar coordinates as the two coordinates σ^1, σ^2. We take the z-axis to be the axis of rotation of the surface and so the origin is a point on this axis. We now have $\underline{\sigma} = (\sigma^1 = r, \sigma^2 = \phi)$.

Any point P on the surface can then be written as

$$\vec{X}(\underline{\sigma}) = \begin{pmatrix} x(r,\phi) \\ y(r,\phi) \\ z(r,\phi) \end{pmatrix} = \begin{pmatrix} r\cos(\phi) \\ r\sin(\phi) \\ z(r) \end{pmatrix}.$$

The z-coordinate is not dependent on ϕ due to rotational symmetry.

This gives us the following tangent vectors

$$\vec{t_1} = \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ z_r \end{pmatrix}, \quad \vec{t_2} = \begin{pmatrix} -r\sin(\phi) \\ r\cos(\phi) \\ 0 \end{pmatrix}.$$

Here we use the subscript to denote partial derivatives, so $z_r = \frac{\partial}{\partial r} z$ and $z_{rr} = \frac{\partial}{\partial r}\frac{\partial}{\partial r} z$.

The metric tensor and the inverse metric tensor then are

$$g_{ij} = \begin{pmatrix} z_r^2 + 1 & 0 \\ 0 & r^2 \end{pmatrix} \quad , g^{ij} = \begin{pmatrix} \frac{1}{z_r^2+1} & 0 \\ 0 & \frac{1}{r^2} \end{pmatrix}.$$

With this we can calculate the Christoffel symbol needed to calculate the extrinsic curvature tensors. These are as follows:

$$\begin{aligned}
\Gamma^1_{11}(\underline{\sigma}) &= \frac{1}{2}\frac{1}{(z_r^2+1)} \cdot \frac{\partial}{\partial r}(z_r^2+1) \\
&= \frac{z_r z_{rr}}{(z_r^2+1)} \\
\Gamma^1_{22}(\underline{\sigma}) &= \frac{1}{2}\frac{1}{(z_r^2+1)} \cdot \left(-\frac{\partial}{\partial r} r^2\right) \\
&= \frac{-r}{(z_r^2+1)} \\
\Gamma^1_{12}(\underline{\sigma}) = \Gamma^1_{21}(\underline{\sigma}) &= 0 \\
\Gamma^2_{11}(\underline{\sigma}) = \Gamma^2_{22}(\underline{\sigma}) &= 0 \\
\Gamma^2_{12}(\underline{\sigma}) = \Gamma^2_{21}(\underline{\sigma}) &= \frac{1}{2}\frac{1}{r^2} \cdot \frac{\partial}{\partial r} r^2 \\
&= \frac{1}{r}
\end{aligned}$$

From equation (2.4) and (2.5) we know that the extrinsic curvature tensors are given by

$$\vec{K_{ij}} = D_i \vec{t_j} = \frac{\partial \vec{t_j}}{\partial \sigma^i} - \Gamma^1_{ij}\vec{t_1} - \Gamma^2_{ij}\vec{t_2}.$$

Calculating for all possible i and j gives us:

$$\begin{aligned}
\vec{K_{11}} &= \frac{\partial \vec{t_1}}{\partial r} - \frac{z_r z_{rr}}{(z_r^2+1)}\vec{t_1} = \begin{pmatrix} 0 \\ 0 \\ z_{rr} \end{pmatrix} - \frac{z_r z_{rr}}{(z_r^2+1)} \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ z_r \end{pmatrix} \\
&= \begin{pmatrix} -\cos(\phi)\frac{z_r z_{rr}}{(z_r^2+1)} \\ -\sin(\phi)\frac{z_r z_{rr}}{(z_r^2+1)} \\ z_{rr} - \frac{z_r^2 z_{rr}}{(z_r^2+1)} \end{pmatrix} \\
\vec{K_{12}} &= \frac{\partial \vec{t_2}}{\partial r} - \frac{1}{r}\vec{t_2} = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} - \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} = \vec{0} \\
\vec{K_{21}} &= \frac{\partial \vec{t_1}}{\partial \phi} - \frac{1}{r}\vec{t_2} = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} - \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} = \vec{0} \\
\vec{K_{22}} &= \frac{\partial \vec{t_1}}{\partial r} - \frac{z_r z_{rr}}{(z_r^2+1)}\vec{t_1} = \begin{pmatrix} -r\cos(\phi) \\ -r\sin(\phi) \\ 0 \end{pmatrix} - \frac{r}{(z_r^2+1)} \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ z_r \end{pmatrix} \\
&= \begin{pmatrix} -r\cos(\phi)\frac{z_r^2}{(z_r^2+1)} \\ -r\sin(\phi)\frac{z_r^2}{(z_r^2+1)} \\ r\frac{z_r}{(z_r^2+1)} \end{pmatrix}
\end{aligned}$$

Now we only need to determine the second fundamental form of Gauss, K_{ij}, to be able to calculate the mean curvatures. From equation (2.6) we immediately find that $K_{12} = K_{21} = 0$. Taking the innerproduct between $\vec{K_{ii}}$ and both sides of equation (2.6) for $i = 1, 2$, we get $\vec{K_{ii}} \cdot \vec{K_{ii}} = K_{ii}^2$.

Calculating the inproducts on the righthandside of the previous equation gives us

$$\begin{aligned}
\vec{K_{11}} \cdot \vec{K_{11}} &= \left(\frac{z_r z_{rr}}{(z_r^2+1)}\right)^2 + \left(z_{rr} - \frac{z_r^2 z_{rr}}{(z_r^2+1)}\right)^2 \\
&= \frac{z_{rr}^2}{(z_r^2+1)} \\
\vec{K_{22}} \cdot \vec{K_{22}} &= \left(\frac{r z_r^2}{(z_r^2+1)}\right)^2 + \left(\frac{r z_r}{(z_r^2+1)}\right)^2 \\
&= \frac{r^2 z_r^2}{(z_r^2+1)}.
\end{aligned}$$

So the second fundamental form of Gauss could be written in matrix form as

$$K_{ij} = \begin{pmatrix} \frac{z_{rr}}{(z_r^2+1)^{\frac{1}{2}}} & 0 \\ 0 & \frac{r z_r}{(z_r^2+1)^{\frac{1}{2}}} \end{pmatrix}$$

And thus the mean curvature H of this case for surfaces with rotational symmetry is given by:

$$H = \frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \frac{1}{2}\sum_i \sum_j g^{ij} K_{ij} = \frac{1}{2}\left(\frac{z_{rr}}{(z_r^2+1)^{\frac{3}{2}}} + \frac{z_r}{r(z_r^2+1)^{\frac{1}{2}}}\right) \tag{2.7}$$

This agrees with the formula used for $\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$ in [8] and [1].

2.4 The shape of a bubble pressed up against a flat plate

Let us now combine all we have seen above in the Young-Laplace's law for our bubble presed up against a flat plate. Since we have rotational symmetry we can rewrite the curvature $\frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$ and thus $\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$ using equation (2.7). But first we need to define the meaning of the function $z(r)$.

When we look at a bubble pressed up against a flat plate, we can see it as a curved line rotating around its symmetry axis. We take the axis to be the z- axis and define $z(r)$ to be the curved line starting from the bottom of the bubble to the top. So at $z = 0, r = 0$ we have the bottom part of the bubble (See figure 4).

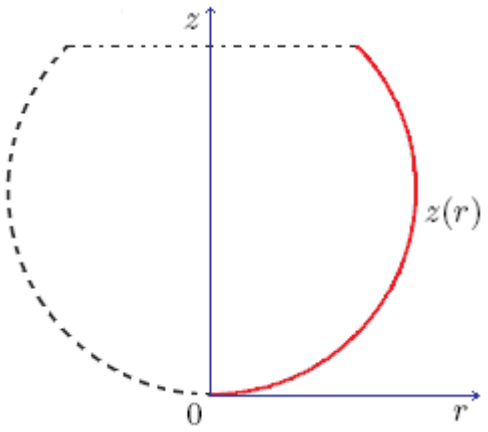

Figure 4: Front view of a single bubble pressed up against a flat plate showing the way $z(r)$ is defined for the use of the Young-Laplace equation.

We can rewrite the Young-Laplace equation to:

$$\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \frac{p_{in} - p_{out}}{\sigma}$$

Let us say that the air in the bubble has density ρ_{in} and the fluid outside of the bubble has density ρ_{out}, both not necessarily uniform. If we take the reference height for the pressure to be $z = 0$ then we can rewrite the right hand side of the equation above (using equation (2.2)) to:

$$\begin{aligned} \frac{p_{in} - p_{out}}{\sigma} &= \frac{1}{\sigma}(p_{in}(z=0) - \rho_{in} g z - p_{out}(z=0) + \rho_{out} g z) \\ &= \frac{\Delta p(z=0)}{\sigma} - \frac{\Delta \rho g z}{\sigma} \end{aligned}$$

We will rewrite the constant $\frac{\Delta \rho g}{\sigma} = {l_c}^{-2}$ as c.

If we look at the bottom of the bubble, we can say that the bubble is spherical there with a certain radius R_b. So we can say that it has curvature $\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \frac{2}{R_b}$ at $z = 0$. This means that, using the Young-Laplace equation, we can write $\frac{\Delta p(z=0)}{\sigma} = \frac{2}{R_b}$.

Combining everything we have so far gives us the equation:

$$\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \frac{z_{rr}}{(z_r^2+1)^{\frac{3}{2}}} + \frac{z_r}{r(z_r^2+1)^{\frac{1}{2}}} = \frac{2}{R_b} - cz = \frac{\Delta p}{\sigma}$$

3 Approximations

In section 2.4 we have seen that the shape of the bubble should obey the following equation

$$\frac{z_{rr}}{(1+z_r^2)^{\frac{3}{2}}} + \frac{z_r}{r(1+z_r^2)^{\frac{1}{2}}} = \frac{2}{R_b} - cz \tag{3.1}$$

where R_b is the radius of curvature at the bottom of the bubble and $c = \frac{\Delta\rho g}{\sigma}$. This is a second order differential equation which could only be solved analytically for some very simple cases, for example when there is no gravity. These simple cases and their solutions can be found in [8].
In this chapter we will provide you some approximations that have been done so far in order to say something about the bubble pressed up against a flat plate by buoyant force. We will look at the two limits for the size of the bubbles, namely very small bubbles and very large bubbles.

3.1 Approximations for very small bubbles

For very small bubbles (size $R_0 \ll l_c$ = the characteristic length), the bubbles are considered so small that we can neglect the pressure drop due to gravity and thus assume the pressure inside and outside of the bubble to be constant over the whole bubble surface. Due to this assumption a bubble would take on the shape of a sphere. Since we push our bubble up against a plate we approximate the shape of that bubble, with size R_0, to be that of a truncated sphere with the same radius R_0 (See figure 5). This is a sphere with the top part of thickness δ cut off. We define δ as the deformation of the bubble.

In the introduction we said that the relation between the force F applied to the bubble and the deformation δ of the bubble is approximated to be a linear relation. Here we will show the approximation yielding the result.

From the Young-Laplace's law we get the following equation:

$$p_{in} = p_{out} + \sigma\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{3.2}$$

Since we approximate the shape of the bubble to be a truncated sphere of radius R_0, we get that the curvature becomes $\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \frac{2}{R_0}$ on every part of the bubble except on the flat contact-area. We also have from chapter 2.2 that $p_{out} = p_l$ on this part of the bubble. At the flat contact-area the curvature becomes $\left(\frac{1}{R_1} + \frac{1}{R_2}\right) = \lim_{R\to\infty} \frac{2}{R} = 0$ while $p_{out} = p_l + \prod$ (See figure 6). Putting this information into equation (3.2) for both cases gives us:

$$\begin{aligned} p_{in} &= p_{out} + \sigma\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \\ &= p_l + \frac{2\sigma}{R_0} \\ &= p_l + \prod + 0. \end{aligned}$$

So in this approximation we obtain $\prod = \frac{2\sigma}{R_0}$. Inserting the former equality in equation (2.3) we get:

$$F = \pi r_c^2 \frac{2\sigma}{R_0} \tag{3.3}$$

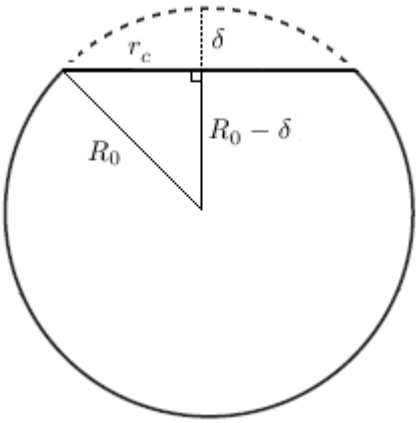

Figure 5: Front view of truncated sphere of radius R_0. The top part which got cut off has thickness δ with a circular base of radius r_c.

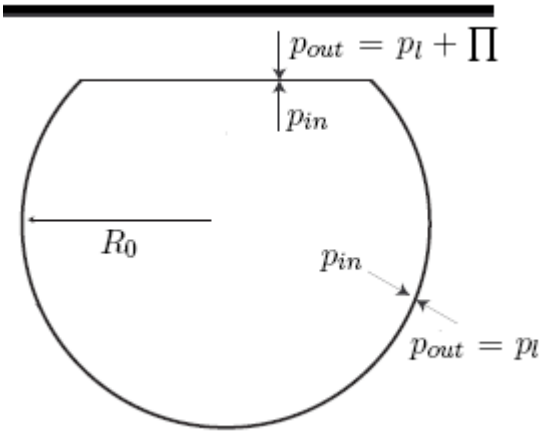

Figure 6: Front view of a bubble pressed up against a flat plate approximated as a truncated sphere of radius R_0. Here we see the two places on the surface where we have applied the Young-Laplace law at.

Furthermore, from the truncated sphere model with deformation δ and the theorem of Pythagoras (See figure 5) we get:

$$\begin{aligned} r_c^2 &= R_0^2 - (R_0 - \delta)^2 \\ &= 2R_0\delta - \delta^2 \end{aligned}$$

Substituting this result in equation (3.3) gives us:

$$F = \pi 2R_0\delta\frac{2\sigma}{R_0} + \mathcal{O}(\delta^2) = 4\pi\sigma\delta + \mathcal{O}(\delta^2).$$

So for $\delta \ll 1$ we get, to leading order, the linear relation between the force and the deformation which has been stated in chapter 1.

Apart from the force and deformation relation we can also look at the value of the radius of the contact-area at the flat plate r_c. For very small bubbles with radius $R_0 \ll l_c$ we know that the volume V of the bubble is given by $V = \frac{4}{3}\pi R_0^3$. From the equations (2.1) and (3.3) we thus get:

$$F = \frac{4}{3}\pi R_0^3\Delta\rho g = \pi r_c^2\frac{2\sigma}{R_0}.$$

Isolating r_c from the previous equation gives

$$r_c = \sqrt{\frac{2}{3}}\frac{R_0^2}{l_c}.$$

Let us now look at equation (3.1). If we again approximate the shape of a bubble to be a sphere of radius $R_0 = R$ we will have that $R_b = R$. So we have

$$\frac{z_{rr}}{(1+z_r^2)^{\frac{3}{2}}} + \frac{z_r}{r(1+z_r^2)^{\frac{1}{2}}} = \frac{2}{R} - cz \tag{3.4}$$

For the shape z we approximate it to be a circular curve with some error. Let us say

$$z = z_0 + \epsilon z_1 + \mathcal{O}\left(\epsilon^2\right)$$

where ϵ is a small parameter and z_0 the circular curve.

$$z_0 = R \pm \sqrt{R^2 - r^2}$$

Before we try to find out what the main error z_1 is, we first rescale the formulas by the radius of the sphere R. We do that by defining

$$z(r) = Ru(x) \text{ and } r = Rx.$$

With this we get:

$$z_{rr} = \frac{1}{R} u_{xx} \text{ and } z_r = u_x. \tag{3.5}$$

Similarly for u as for z, we can write

$$u = u_0 + \epsilon u_1 + \mathcal{O}\left(\epsilon^2\right)$$

with $\epsilon \ll 1$, the same small parameter as before, and

$$u_0 = \frac{1}{R} z_0 = 1 \pm \sqrt{1 - x^2}.$$

By applying all the rescalings to equation (3.4) and multiplying the whole equation by R we get the differential equation for the the rescaled shape u.

$$\frac{u_{xx}}{(1 + u_x^2)^{\frac{3}{2}}} + \frac{u_x}{x(1 + u_x^2)^{\frac{1}{2}}} = 2 - cR^2 u \tag{3.6}$$

Let us substitute $u = u_0 + \epsilon u_1 + \mathcal{O}\left(\epsilon^2\right)$ in equation (3.6) and only look at terms up to order ϵ. We will do this in short steps starting with by rewriting the two denominators in the equation. Let $a \in \{1, 3\}$ then

$$\begin{aligned}
\left(1 + u_x^2\right)^{-\frac{a}{2}} &= \left(1 + u_{0x}^2 + \epsilon 2 u_{0x} u_{1x} + \mathcal{O}\left(\epsilon^2\right)\right)^{-\frac{a}{2}} \\
&= \left(1 + u_{0x}^2\right)^{-\frac{a}{2}} \left(1 + \epsilon \frac{2 u_{0x} u_{1x}}{\left(1 + u_{0x}^2\right)} + \mathcal{O}\left(\epsilon^2\right)\right)^{-\frac{a}{2}}
\end{aligned}$$

Taking the Taylor expansion of the previous equation at $\epsilon = 0$ gives

$$\left(1 + u_x^2\right)^{-\frac{a}{2}} = \left(1 + u_{0x}^2\right)^{-\frac{a}{2}} \left(1 - \epsilon \frac{a u_{0x} u_{1x}}{\left(1 + u_{0x}^2\right)} + \mathcal{O}\left(\epsilon^2\right)\right)$$

With this we get

$$\begin{aligned}
\frac{u_{xx}}{(1 + u_x^2)^{-\frac{3}{2}}} &= \left[u_{0xx} + \epsilon u_{1xx} + \mathcal{O}\left(\epsilon^2\right)\right] \left[\left(1 + u_{0x}^2\right)^{-\frac{3}{2}} \left(1 - \epsilon \frac{3 u_{0x} u_{1x}}{\left(1 + u_{0x}^2\right)} + \mathcal{O}\left(\epsilon^2\right)\right)\right] \\
&= \frac{u_{0xx}}{\left(1 + u_{0x}^2\right)^{\frac{3}{2}}} + \epsilon \left(\frac{u_{1xx}}{\left(1 + u_{0x}^2\right)^{\frac{3}{2}}} - \frac{3 u_{0x} u_{0xx} u_{1x}}{\left(1 + u_{0x}^2\right)^{\frac{5}{2}}}\right) + \mathcal{O}\left(\epsilon^2\right)
\end{aligned}$$

$$\begin{aligned}\frac{u_x}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}} &= \frac{1}{x}\left[u_{0x}+\epsilon u_{1x}+\mathcal{O}\left(\epsilon^2\right)\right]\left[\left(1+u_{0x}^2\right)^{-\frac{1}{2}}\left(1-\epsilon\frac{u_{0x}u_{1x}}{\left(1+u_{0x}^2\right)}+\mathcal{O}\left(\epsilon^2\right)\right)\right] \\ &= \frac{u_{0x}}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}}+\epsilon\left(\frac{u_{1x}}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}}-\frac{u_{0x}^2u_{1x}}{x\left(1+u_{0x}^2\right)^{\frac{3}{2}}}\right)+\mathcal{O}\left(\epsilon^2\right)\end{aligned}$$

So equation (3.6) becomes

$$\begin{aligned}2-cR^2(u_0+\epsilon u_1) &= \frac{u_{0xx}}{\left(1+u_{0x}^2\right)^{\frac{3}{2}}}+\frac{u_{0x}}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}} \\ &\quad +\epsilon\left(\frac{1}{\left(1+u_{0x}^2\right)^{\frac{3}{2}}}u_{1xx}+\left(\frac{1}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}}-\frac{3u_{0x}u_{0xx}}{\left(1+u_{0x}^2\right)^{\frac{5}{2}}}-\frac{u_{0x}^2}{x\left(1+u_{0x}^2\right)^{\frac{3}{2}}}\right)u_{1x}\right) \\ &\quad +\mathcal{O}\left(\epsilon^2\right)\end{aligned}$$

Note that for $u_0 = 1-\sqrt{1-x^2}$ we have that

$$\frac{u_{0xx}}{(1+u_{0x}^2)^{\frac{3}{2}}}+\frac{u_{0x}}{x(1+u_{0x}^2)^{\frac{1}{2}}}=2.$$

So we are left with the following equation:

$$\begin{aligned}-cR^2(u_0+\epsilon u_1) &= \epsilon\left(\frac{1}{\left(1+u_{0x}^2\right)^{\frac{3}{2}}}u_{1xx}+\left(\frac{1}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}}-\frac{3u_{0x}u_{0xx}}{\left(1+u_{0x}^2\right)^{\frac{5}{2}}}-\frac{u_{0x}^2}{x\left(1+u_{0x}^2\right)^{\frac{3}{2}}}\right)u_{1x}\right) \\ &\quad +\mathcal{O}\left(\epsilon^2\right)\end{aligned}$$

From the equation above we see that our small parameter ϵ equals R^2. So we have:

$$\begin{aligned}-\epsilon cu_0 &= \epsilon\left(\frac{1}{\left(1+u_{0x}^2\right)^{\frac{3}{2}}}u_{1xx}+\left(\frac{1}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}}-\frac{3u_{0x}u_{0xx}}{\left(1+u_{0x}^2\right)^{\frac{5}{2}}}-\frac{u_{0x}^2}{x\left(1+u_{0x}^2\right)^{\frac{3}{2}}}\right)u_{1x}\right) \\ &\quad +\mathcal{O}\left(\epsilon^2\right)\end{aligned}$$

We will solve this equation for the order ϵ terms. So we have to solve the equation:

$$f(x)u_{1xx}+g(x)u_{1x}=h(x) \tag{3.7}$$

with

$$\begin{aligned}f(x) &= \frac{1}{\left(1+u_{0x}^2\right)^{\frac{3}{2}}} \\ &= \left(1-x^2\right)^{\frac{3}{2}} \\ g(x) &= \left(\frac{1}{x\left(1+u_{0x}^2\right)^{\frac{1}{2}}}-\frac{3u_{0x}u_{0xx}}{\left(1+u_{0x}^2\right)^{\frac{5}{2}}}-\frac{u_{0x}^2}{x\left(1+u_{0x}^2\right)^{\frac{3}{2}}}\right) \\ &= \left(1-x^2\right)^{\frac{3}{2}}\left(\frac{1}{x\left(1-x^2\right)}-\frac{3x}{\left(1-x^2\right)^6}-\frac{x}{\left(1-x^2\right)}\right) \\ &= \left(1-x^2\right)^{\frac{3}{2}}\left(\frac{1}{x}-\frac{3x}{\left(1-x^2\right)^6}\right)\end{aligned}$$

$$\begin{aligned} h(x) &= -cu_0 \\ &= -c\left(1-\left(1-x^2\right)^{\frac{1}{2}}\right) \end{aligned}$$

Dividing equation (3.7) by $f(x)$ for $x \neq 1$ gives

$$\begin{aligned} u_{1xx} + \frac{g(x)}{f(x)} u_{1x} &= \frac{h(x)}{f(x)} \quad \text{which is} \\ u_{1xx} + \left(\frac{1}{x} - \frac{3x}{\left(1-x^2\right)^6}\right) u_{1x} &= -c\left(\frac{1}{\left(1-x^2\right)^{\frac{3}{2}}} - \frac{1}{\left(1-x^2\right)}\right) \end{aligned}$$

Let us now introduce $v = u_{1x}$. The previous equation then becomes

$$\frac{dv}{dx} + \left(\frac{1}{x} - \frac{3x}{\left(1-x^2\right)^6}\right) v = -c\left(\frac{1}{\left(1-x^2\right)^{\frac{3}{2}}} - \frac{1}{\left(1-x^2\right)}\right) \tag{3.8}$$

To solve equation (3.8) we first look for the homogeneous solution. So we have

$$\frac{dv}{dx} + \left(\frac{1}{x} - \frac{3x}{\left(1-x^2\right)^6}\right) v = 0.$$

This gives us the following:

$$\begin{aligned} \frac{dv}{dx} &= -\left(\frac{1}{x} - \frac{3x}{\left(1-x^2\right)^6}\right) v \\ \frac{1}{v}\frac{dv}{dx} &= -\left(\frac{1}{x} - \frac{3x}{\left(1-x^2\right)^6}\right) \\ \frac{d\ln(|v|)}{dx} &= -\left(\frac{1}{x} - \frac{3x}{\left(1-x^2\right)^6}\right) \\ \ln(|v|) &= -\ln(|x|) + \frac{3}{10\left(1-x^2\right)^5} + c_1 \\ v &= \frac{c_2}{x} e^{\frac{3}{10\left(1-x^2\right)^5}} \end{aligned}$$

Where c_1 and c_2 are arbitrary constants.

So the homogeneous solution $v_h(x)$ of equation (3.8) satisfies $v_h(x) = \frac{c_2}{x} e^{\frac{3}{10\left(1-x^2\right)^5}}$.

Let us now assume that the solution to equation (3.8) is of the form $v(x) = b(x)v_h(x)$ with $v_h(x) = \frac{1}{x}e^{\frac{3}{10(1-x^2)^5}}$. Substituting $v(x)$ in equation (3.8) gives us

$$\begin{aligned} b(x) &= \int_0^x -c\left(\frac{1}{(1-y^2)^{\frac{3}{2}}} - \frac{1}{(1-y^2)}\right)\frac{1}{v_h(y)}dy. \\ &= \int_0^x -c\left(\frac{1}{(1-y^2)^{\frac{3}{2}}} - \frac{1}{(1-y^2)}\right)ye^{-\frac{3}{10(1-y^2)^5}}dy \end{aligned}$$

With this we have found the solution to equation (3.8) to be

$$v(x) = \left(\frac{1}{x}e^{\frac{3}{10(1-x^2)^5}}\right)\int_0^x -c\left(\frac{1}{(1-y^2)^{\frac{3}{2}}} - \frac{1}{(1-y^2)}\right)ye^{-\frac{3}{10(1-y^2)^5}}dy.$$

We have defined $u_{1x} = v$. So we have found the formula of the error of the approximation u_1 to be

$$\begin{aligned} u_1(x) &= \int_0^x v(z)dz \\ &= \int_0^x \left(\frac{1}{z}e^{\frac{3}{10(1-z^2)^5}}\right)\int_0^z -c\left(\frac{1}{(1-y^2)^{\frac{3}{2}}} - \frac{1}{(1-y^2)}\right)ye^{-\frac{3}{10(1-y^2)^5}}dydz \end{aligned}$$

with $c = \frac{\Delta\rho g}{\sigma}$.

3.2 Approximations for very large bubbles

For very large bubbles ($R_0 \gg l_c$) there is an approximation in [5] that assumes the shape of the bubble to be nearly a cylinder. Even though the shape of a bubble in reality is not at all a cylinder we would still like to include this approximation for larger bubbles.

Just like in the case of the smaller bubbles, we will look at the value of the contact-radius r_c. We assume that the pushed bubble has the shape of a pancake, meaning a cylinder of radius r_c, but instead of the flat edge we have circular edge of a certain radius R. So this means the pancake has a thickness $2R$ (See figure 7).

Let us look at the value of R. We assume that the pressure inside the bubble is constant over the whole volume. Now let us look as the pressure drop outside of the bubble. Take the bottom of the bubble to be a reference height. Because the curvature goes to zero at the bottom of the bubble, we find from Young-Laplace's law that the pressure outside of the bubble at reference height $p_{out}(0)$ must be equal to the pressure inside of the bubble $p_{in}(0)$. If we look at the surface halfway of the bubble, so at depth $z = -R$ from the reference height, we see that the radii of curvature are $R_1 \gg 1$ and $R_2 = R$. From the Young-Laplace law at that point we thus get $\Delta p = p_{in}(-R) - p_{out}(-R) = \frac{\sigma}{R}$.

So combining these facts with equation (2.2) we get:

$$\begin{aligned} p_{out}(-R) &= p_{in}(-R) - \frac{\sigma}{R} \\ &= p_{out}(0) - \rho g R \\ &= p_{in}(0) - \rho g R \end{aligned}$$

If we consider the pressure inside the bubble to be uniform ($p_{in}(-R) = p_{in}(0)$), we get $R = \sqrt{\frac{\sigma}{\rho g}}$, which is the characteristic length l_c if we neglect the density of the gas inside the bubble.

We can thus approximate the volume of the pancake to be a cylinder of height $2l_c$ with the contact-area as its base. By conservation of the volume we thus get

$$2\pi r_c^2 l_c = \frac{4}{3}\pi R_0^3$$

from which follows

$$r_c = \sqrt{\frac{2}{3}}\frac{R_0^{\frac{3}{2}}}{\sqrt{l_c}}.$$

In reality the bubble does not take the pancake form but gets curvier the closer the surface is to the flat plate (See figure 8).

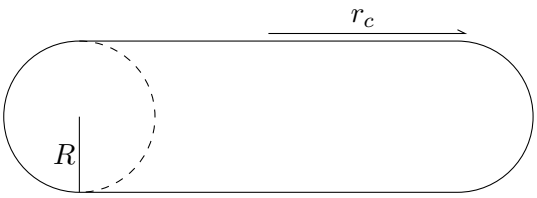

Figure 7: Front view of the approximation of the shape of a large bubble pressed up against a flat plate. Here r_c is the radius of the circular contact-area and R the radius of the circular edge of the pancake.

Figure 8: Front view of the actual shape of a large bubble pressed up against a flat plate.

4 Simulations with the Surface Evolver

Now that we have some approximations, we would like to see if they make any sense. But more than that, we are also interested in the intermediate region for which we do not have any approximations yet. For this purpose we have decided to run simulations with a computer program called Surface Evolver. The Surface Evolver is a program created by the Kenneth A. Brakke that minimizes the energy of a surface subject to constraints. In the program the surface is approximated by vertices, edges and triangular facets. One first starts with some of those predefined in a data-sheet used for the simulation. In each simulation step, the program would determine the forces on each vertex and move it according to the forces. The user also has the option to refine the surface, which would cause the program to split each facet into smaller triangles and therefore adding new edges and vertices to the simulation.

In this section we will first tell something about the program and how to operate Surface Evolver. This will be followed by details and results of the two simulation experiments which we have done. Since we don't know how accurate the results from simulations with Surface Evolver are, we have experimented with it to see how well our configuration converges to the reality as function of the number of iterations taken with Surface Evolver. We have also looked at how our configuration converges to reality as function of the amount of vertices.

4.1 Operating Surface Evolver

We would like to use the Surface Evolver to simulate a bubble pressed up against a flat plate. In order to do that we must first define some vertices, edges, facets and bodies in a data-sheet used by the Surface Evolver. A data-sheet is just a file with extension *.fe* that can be written and read in simple text making programs such as notepad. In the data-sheet a vertex is defined by its spatial coordinates. An edge, which is directed, will be defined by the vertices it connects. The facets are defined by the oriented edge loop which gives the border of the facet. And a body is defined by the facets on its boundary. The starting facets may be of any shape, for the Surface Evolver will cut it in triangles automatically. In doing so Surface Evolver would add new vertices, edges and facets during the simulation. For the fashion in which this has to be done, we suggest you to take a look at the data-sheet included as an appendix or at examples provided with the program and in the manual [2].

If we start out with a cube on the x-y plane without any forces applied on it, the Surface Evolver will evolve it into a sphere. If we add gravity to it, it will get flattened against the x-y plane. This is almost like the bubble we want to simulate. However, Surface Evolver does not take care of the facets on the x-y plane in the way that would give us a realistic bubble shape. It leaves those facets there where it was. From experience we know that a bubble which is pushed softly against a flat plane would have a very small contact-area. This could not be attained if we use the whole cube as initial configuration. So we have to remove the facet which is already touching the plane.

After running some simulations with the cube-like figure as initial configuration, we find out we can help Surface Evolver a bit in the process of evolving by creating an initial configuration that is shaped more like what it needs to evolve into. This gave us the idea to have the following two shapes as initial configuration (See figure 9 and 10).

Here configuration $p1$ (figure 9) will be used for small bubbles and configuration $p2$ (figure 10) for larger bubbles.

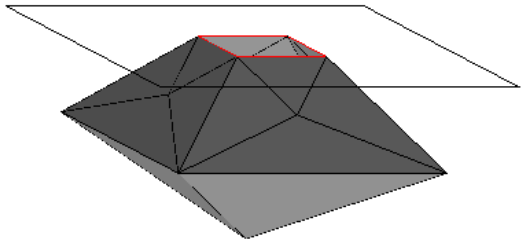

Figure 9: Initial configuration p1, for the simulation of a small bubble pressed up against a flat plate in Surface Evolver.

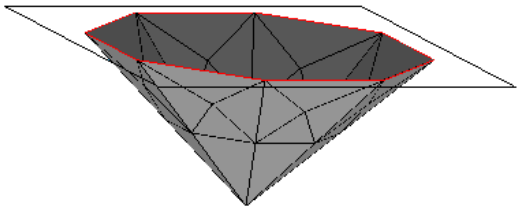

Figure 10: Initial configuration p2, for the simulation of a large bubble pressed up against a flat plate in Surface Evolver.

4.1.1 Constraints, energies and other integrals

By removing the contact-area one also removes the physical properties of the area from the simulator. This needs to be corrected by the use of constraints and predefined energy at the removed area in the data-sheet. Any other physical property, such as not bieng able to rise above the flat plate, should also be provided as a constraint as follows:

constraint n NONNEGATIVE GLOBAL
function: x3=0

Here n is an integer used as a name for the constraint.

Now we still need to give the simulation the information that there is also surface tension energy on the removed contact area. Since we do not have any vertices, edges or facets in the contact-area, we can only give information about the area by the use of its edge. This reminds us of Stokes' Theorem:

$$\int\int_S curlF \cdot \hat{N} dS = \oint_C F \cdot dr$$

We know that the energy at the removed area is given by

$$\sigma \cdot \text{Area} = \sigma \int\int_S 1 dS = \int\int_S \sigma \hat{k} \cdot \hat{N} dS.$$

We would like to have a vector F to satisfy $curlF = \sigma\hat{k}$. One finds that either $\sigma x\hat{j}$ or $-\sigma y\hat{i}$ does the trick. So, taking the second one, this gives us

$$\sigma \cdot \text{Area} = \oint_C -\sigma y\hat{i} \cdot dr.$$

This information can be given to Surface Evolver by the inclusion of the following constraint in the data-sheet:

constraint n
formula: x3 = 0
energy:
e1: -(σ*y)
e2: 0
e3: 0

In the same way as with the energy, we see that we can use Stokes' theorem to get information about the contact-area of the bubble, which happens to be the removed area in the simulation. We only have to delete the constant σ in the formula's above. The commands needed to actually get the value of the contact-area will be provided in the next part.

4.1.2 Commands

Once we have opened the Surface Evolver we first need to load the data-sheet which we would like to use. This can be done by typing the path and filename without its extension. For example, if your data-sheet is called *cube.fe* and it is located in the map *datasheets* that is located in the same map as the program Surface Evolver, type *datasheets/cube* and press enter. Then the data-sheet has been loaded.

To operate Surface Evolver we have to type in commands and press enter. Surface Evolver has a wide set of build-in commands from which we will show some so you will get the idea. For all the commands we refer you to the Surface Evolver manual [2]. By typing *"h"* you will see a list some of the possible commands. the command *"s"* prompts up a new window in which the user can see the shape defined in the data-sheet. It also leaves the user in the graphic command mode. To leave this mode (or to leave anything at all) use the command *"x"* or *"q"*.

To actually let the Surface Evolver start to evolve our surface we use the command *"g"*. This lets the Surface Evolver do one step with the gradient descent method. To get more steps in one go, one has to add the ammount of steps after the command used. For example, the command *"g10"* lets the Surface Evolver do ten steps with the gradient descent method. To refine the configuration, use the command *"r"*. These two commands alone can get you quite far in the process of simulating a bubble. However, vertices and edges would eventually run into each other. This will prevent the Surface Evolver from achieving a natural state of minimum energy. So we have to prevent the vertices and edges to run into each other. This can be done by using some of the re-gridding commands which are built into Surface Evolver. We have listed these commands and other useful commands in table 1.

Command	Description	Command	Description
h	Trigger help menu.	r	Refine triangulation.
x,q	Quit.	t	Remove tiny edges.
s	Prompt graphic mode.	u	Equianulate.
g	Take a step with the gradient descent method.	l	Subdivide long edges.
		K	Skinny triangle long edge divide.

Table 1: Useful Surface Evolver commands

If the user would like to do more actions in one commandline, one can type the commands in a bracket separated by semicolons. For example the command *"{g; r; g; r;}"* is the same as typing *"g"* then *"r"* then *"g"* and then *"r"* separately. Note that the command *"{g; r;}2"* yields the same effect. One can also create ones own shortcuts for commands. This can be done by first entering the name of your command, then *":="* followed by the command the name should represent. For example *"gr:={g; r;}"* gives us the new command *"gr"* which does the same as *"{g;r;}"*. This command can be used until one redefines it or until one quits Surface Evolver.

Redefining commands every time after starting up the simulation could be tiring. For that sake we can have some shortcuts defined directly after loading the data-sheet by just adding the command *"read"* in the data-sheet followed by anything you would have otherwise done after the data-sheet has been loaded (except the command *"s"* for this will prompt the graphic command mode and stop all the normal commands).

To actually collect data from Surface Evolver digitally we can use the commands *"printf "content" >> "filename.txt""* to let Surface Evolver create a file (filename.txt) in which the information (content) we need gets recorded.

For all the other tricks we have used, we refer the reader to the data-sheet included as an appendix.

4.1.3 Converging simulation data to reality

The surface Evolver works with dimensionless units, but we would like to relate the information we get from the simulation to the situation in reality. For this we need to know how to convert dimensionless units in the program to the quantities in reality which have dimensions. The physical units which come into play in our problem are: The surface tension σ with unit N/m, the gravity constant g with unit N/kg, the density difference of the two substances $\Delta\rho$ with unit kg/m^3 and the size of the bubble R with unit m.
We then see that the following quantity is dimenstionless:

$$\frac{R}{\sqrt{\frac{\sigma}{\Delta\rho g}}} = \frac{R}{l_c}$$

This quantity must be the same in the simulation and in reality. This gives us an equation $\frac{R}{l_c}_{SE} = \frac{R}{l_c}$ in which the subscript SE means values from Surface Evolver. So basically we can see Surface Evolver as a rescaling of the reality. Note that instead of the quantity R we can place any other quantity h that has dimension of length m. So we have $\frac{h}{l_c}_{SE} = \frac{h}{l_c}$ for all length quantities h.

In Surface Evolver the quantities R, $\Delta\rho$ and σ are fixed by the data-sheet to satisfy $\frac{4}{3}\pi R^3 = 1$, $\Delta\rho = 1$ and $\sigma = 1$. With some effort the user of the program can redefine them to other values (see the manual [2] for more information.), but we will keep it this way and only vary the value of the gravity constant G. Since we use $l_c \approx 1.63$mm we thus get for any quantity h of length dimension m that $h = h_{SE} \cdot \sqrt{G} \cdot 1.63$mm.

4.1.4 Acquiring the value of the contact-radius

Surface Evolver should satisfy the laws of physics. So when the bubble is in equilibrium configuration we would have the force balance stated in equation (2.3). The upward force on the bubble in the simulation is the gravitational force G. This forced should be balanced by the disjoining pressure $\prod$ in the thin film between the bubble and the plate multiplied by the contact area A. This disjoining pressure is the same as the pressure inside the bubble p. So we get $G = pA$.

There are two ways to acquire the information about the radius from the simulator:

1. We ask for the area of the contact area A and calculate the radius r_c from the equation $A = \pi r_c^2$. We will call this method the "Area Method".

2. We ask for the pressure p inside of the bubble. With that we calculate the expected contact area A from the equation $G = pA$ and then calculate the radius just like in the previous method. We will call this method the "Pressure Method".

Even though most of the results from the simulation show that values acquired by the Pressure Method converge faster to a final result, we will still provide the results using the Area Method since it gives a better idea of what is going on.

4.2 The simulations: Area Method or Pressure Method? Test for convergence with respect to the amount of iterations

Just like with any other simulation, Surface Evolver gives an approximation of the reality. The error in the approximation would go to zero as the time (the amount of iterations on Surface Evolver) and the number of facets goes to infinity, assuming that the simulation converges and no unrealistic bubble shapes occur. However, we do not have so much time and having too many facets also slows down the simulation a lot. Furthermore, having the simulation running unattended is also not an option. During absence there is no guarantee that the shape of the bubble will actually evolve into a realistic shape and one mistake can drastically change the bubble into some spiked ball. For this reason we have decided to test how Surface Evolver converges with the number of iterations, which will be done in this section, and to test the convergence with the amount of facets, which will be presented in the section 4.3. We will do this by looking at the value of the contact-radius r_c.

In section 4.1.4 we have stated that there are at least two different methods to acquire the value of r_c in Surface Evolver, namely the Pressure Method and the Area Method. We have decided to look at the differences in these two methods as well.

For the simulation of the convergence with respect to the amount of iterations, which we from now on will denote as n, we started out with initial condition p1 (see figure 9) and started the simulation with the command *"goo"* from the data-sheet until the bubble looks decent and the shape of the contact area hardly changes. We did this in order to get the configuration to look more like a bubble and become more stable. Then we refined it and used the command *"gogo 0.005"* from the data-sheet. We let the program continue for over two hours and then we stopped the simulation.

4.2.1 Results

The result of the simulation can be seen in figure 11. We see that the value of r_c for the Pressure Method is steady right from the start while the value of r_c from the Area Method is still decreasing towards the one from the Pressure Method. We hereby conclude that the results from the simulation will be much more accurate if we attain the information by the use of the Pressure Method.

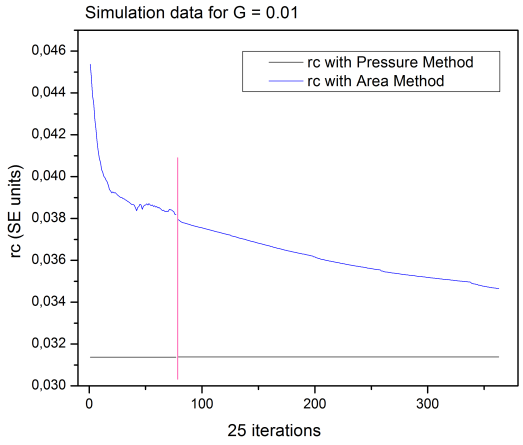

Figure 11: Simulation data for G = 0.01 showing the value of r_c in Surface Evolver using the two different approaches (Pressure Method and Area Method) as a function of the number of iterations. The vertical line in the graph denotes the moment in which we have refined the configuration.

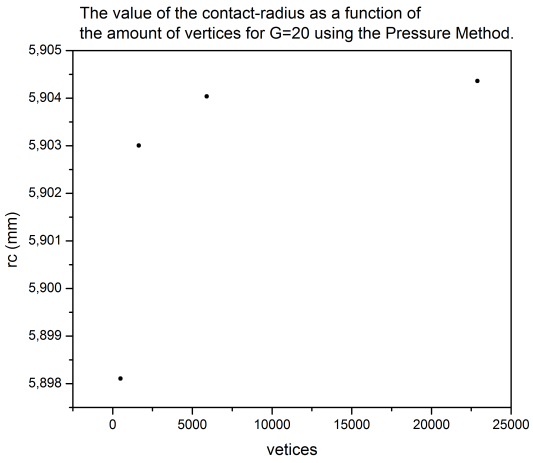

Figure 12: Simulation data for G = 20 showing the value of r_c in Surface Evolver using the Pressure Method as a function of the amount of vertices.

4.3 The simulations: Test for convergence with respect to the amount of vertices

A bubble pressed up against a plate can be represented by an infinite amount of vertices, but as we already said in the previous section, we can not achieve that on Surface Evolver. For this reason we decided to look at how the acquired information from Surface Evolver varies as a function of the finite amount of vertices which we will denote as N. Once again we look at the value of the contact-radius r_c using the two different methods, namely the Pressure Method and the Area Method. So far we have tried this for the following three values of the gravitational constant G: $15, 20$ and 25. For each of the values G we run the simulation starting with initial configuration p2 (see figure 10) until everything looks satisfying, meaning that the contact-area looks circular and smooth and the bubble does not change its shape that much. Then we record the value of the contact-radius together with the amount of vertices and then refine the configuration and continue the simulation with more vertices. Once again, after we feel satisfied we record data and refine the configuration and continue. This eventually leads us to having data of the configurations when they have approximately 400, 1600, 6000 and 23000 vertices. The result of the simulation for $G = 20$ is shown in figure 12

We are curious as to how the value of the contact-radius r_c from Surface Evolver converges as function of the amount of vertices N. If it would converge very fast (for example exponentially), we could stop just stop the simulation at a reasonable value of N. Making N any larger would hardly reduce the error. When one look at the simulation data in figure 12, one sees that it does not converges very fast. And since many approximations of shapes with smaller components, such as approximation a circle by a regular polygon with N edges, converges algebraically with N, we assume that the value of the contact-radius r_c converges algebraically as well. We thus assume

$$r_c(N) = r_{c_\infty} + \frac{\lambda}{N^\gamma}$$

where r_{c_∞} is the unknown limit of the value of the contact-radius, λ is a real number and γ a positive number. In this case, making the value of N a little bit bigger would not really bring us closer to the value of the contact-radius we search for. So we would rather extrapolate the value of r_{c_∞}. We do this in the following manner: First we guess the value of of r_{c_∞} and plot $|r_c(N) - r_{c_\infty}|$ as a function of N on double log-scales. If the guess is quite accurate one will see that the function $|r_c(N) - r_{c_\infty}|$ will be more like a straight line as N gets larger while varying the value of r_{c_∞} a little from the accurate value will bend the straigt line end upwards or downwards. We do this for some guessed values of r_{c_∞}. Once it seems as though we might have found the value of r_{c_∞} (the function $|r_c(N) - r_{c_\infty}|$ looks like a straight line at the end), we try to find out what γ is. We again guess the value of γ first and then plot $\alpha N^{-\gamma}$, where α is a real number to be chosen freely. The value of γ we search for is the value for which the functions $|r_c(N) - r_{c_\infty}|$ and $\alpha N^{-\gamma}$ are parallel.

4.3.1 Results

In figure 13 we see the approach described above being applied on the data for $G = 20$ with Pressure Method. Here we show the reader the function $|r_c(N) - r_{c_\infty}|$ for three guessed values of r_{c_∞}. One sees that the uppermost plot bends upwards as N gets larger while the lowermost plot bends downwards. This suggests that the middle one is closer to the actual limit of $r_c(N)$. We take the value of r_{c_∞} at the middle plot to be the value of the contact-radius. In the same plot we also plot our function $\alpha N^{-\gamma}$ with the value of γ, for which we find that the two functions $|r_c(N) - r_{c_\infty}|$ and $\alpha N^{-\gamma}$ are parallel, as a dashed line.

We also do this for the Area Method and repeat for $G = 15$ and $G = 25$ as well. By doing so we find that the value of γ is approximately $\gamma \approx \frac{1}{2}$ for the Area Method and $\gamma \approx 1$ for the Pressure Method. So we see once again that the Pressure Method gives faster convergence since the value of γ is higher. For this reason we have decided to only plot our last result using data from the Pressure Method. As our last result we have the plot of the approximated value of the contact-radius as the function of the bubble size R (See figure 14).

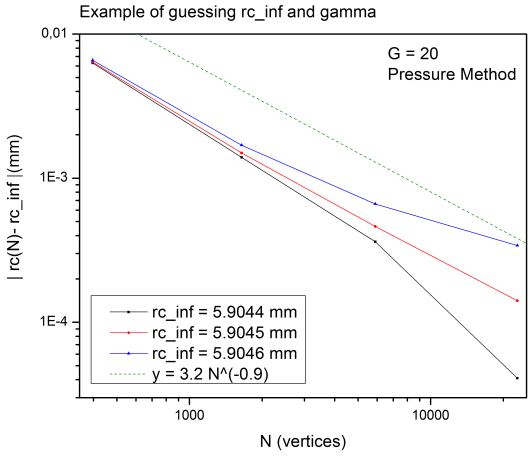

Figure 13: Plot showing the method of guessing the value of r_{c_∞} and γ for G = 20 with the Pressure Method. The three solid lines are the functions $|r_c(N) - r_{c_\infty}|$ for a guessed r_{c_∞} which can be seen in the legend. The dotted line is the function $\alpha N^{-\gamma}$ for appropriate α and γ.

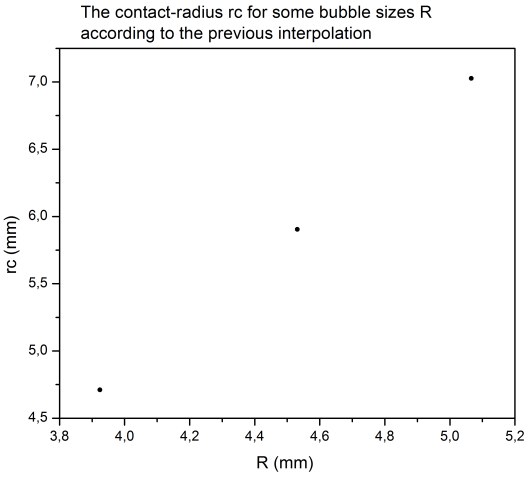

Figure 14: Simulation from Surface Evolver showing the value of the contact-radius r_c as function of the size of the bubble R.

5 Concluding remarks

Considering this subject and the things that we have done so far in this thesis, we see that there is still a lot to be done. We have only roughly found a way to operate Surface Evolver in which we concluded that we should look at the value of the pressure attained from the simulation and interpolate rather than trying to get the amount of iterations and vertices to infinity. Whether or not the data acquired from the simulation agrees tot the approximations is still unknown. The integrals for the formula of the error in the approximation that the shape of small bubbles are spherical is still left as an integral.

So we would like to suggest to future researchers who would like to continue from where we are to look into the simulator a bit more. Some more simulations for many different bubble sizes could be done to check consistency with the approximations. One might also try to look at the integrals, which still need to be evaluated, numerically to get an idea of how accurate the spherical approximation is. Furthermore, we have also found a paper which approximates the shape of a large bubble in a similar way as how we got the differential equation for the shape of a small bubble (see [1, p.228-240]). This could also be investigated.

6 References

[1] Edwin Blaisdell. The physical properties of interfaces of large radius of curvature. In *J. Math. Phys.* Inst. Tech., 1940.

[2] K.A. Brakke. *Surface Evolver manual Version 2.30.*

[3] Franois David. Geometry and field theory of random surfaces and membranes. In *Statistical Mechanics of Membranes and Surfaces*, pages 151–161. World Scientific Pub., 2004.

[4] Timon Idema. *Structure, shape and dynamics of biological membranes.* PhD thesis, Universiteit Leiden, 2009.

[5] Gijs Katgert. *Flow of Foams.* PhD thesis, Universiteit Leden, 2008.

[6] B. Lautrup. *Physics of Continuous Matter.* I0P Publishing Ltd., 2005.

[7] Daniel Miedema. The deformation of a 1d foam. Master's thesis, Universiteit Leiden, 2010.

[8] H.M. Princen. The equilibrium shape of interface drops and bubbles. rigid and deformable particles at interfaces. In Egon Matijevic, editor, *Surface and Colloid Sciences*, volume 2. Wiley-Interscience, 1969.

[9] Murray R. Spiegel. *Theory and problems of vector analysis.* Shaums Outline. McGraww-Hill Book Co., Singapore, 1974.

A Data-sheet for Surface Evolver: Initial configuration p1

```
PARAMETER angle = 180 // interior angle between plane and surface in degrees

gravity_constant 0 // start with gravity off

#define WALLT (-cos(angle*pi/180)) // virtual tension of facet on plane


constraint 1 /* the table top */
formula: x3 = 0
energy: // for contact angle
e1: -(WALLT*y)
e2: 0
e3: 0


constraint 2 NONNEGATIVE GLOBAL
function: x3=0


quantity facet_area info_only method edge_vector_integral global
vector_integrand:
q1: 0
q2: x
q3: 0


vertices
1 -0.25 -0.25 0.0 constraint 1 /* 4 vertices on plane */
2 0.25 -0.25 0.0 constraint 1
3 0.25 0.25 0.0 constraint 1
4 -0.25 0.25 0.0 constraint 1
5 -0.75 -0.75 0.5
6 0.75 -0.75 0.5
7 0.75 0.75 0.5
8 -0.75 0.75 0.5
9 0.0 0.0 1.0


edges /* given by endpoints and attribute */
1 1 2 constraint 1 /* 4 edges on plane */
2 2 3 constraint 1
3 3 4 constraint 1
4 4 1 constraint 1
5 5 6
6 6 7
7 7 8
8 8 5
9 1 5
10 2 6
11 3 7
12 4 8
13 5 9
```

```
14 6 9
15 7 9
16 8 9


faces /* given by oriented edge loop */
1 1 10 -5 -9
2 2 11 -6 -10
3 3 12 -7 -11
4 4 9 -8 -12
5 5 14 -13
6 6 15 -14
7 7 16 - 15
8 8 13 -16


bodies /* one body, defined by its oriented faces */
1 1 2 3 4 5 6 7 8 volume 1 density 1


read
re := {refine edges where on_constraint 1 }

inoutp := {printf "Facet area \t pressure \t # vertices \t p*A \n" >> "p1G001vertex.txt"
}
outp :={printf "%f10 \t %f10 \t %f10 \t %f10 \n",sum (edges where on_constraint 1,facet_area),
body[1].pressure, sum(vertices,1), sum (edges where on_constraint 1,facet_area)*body[1].pressure
>> "p1G001vertex.txt"}

procedure edge_equal (real tt) {
ll:=0;
facet_average :=0;
foreach edge ee do if on_constraint 1 then
{ll:=ll+1; facet_average := facet_average+max(ee.facet,area)};
facet_average := facet_average/ll;
refine edges ee where on_constraint 1 && 1.25*max(ee.facet,area)>facet_average;
K 0.4; u; edgeweed tt; u;
}

procedure gogo (real tt){
{{g25; u; edgeweed tt; u; outp; edge_equal(tt)} 100}
}

goo := {gogo2; gogo2b 5; r; gogo2c 5; U; r; {gogo(0.01)} 5; {gogo(0.005)} 5; {gogo(0.002)}
10; {gogo(0.001)} 10;}

set facet color grey; set edge color red where on_constraint 1;
G 0.01;
```

B Data-sheet for Surface Evolver: Initial configuration p2

```
PARAMETER angle = 180 // interior angle between plane and surface, degrees

gravity_constant 0 // start with gravity off

#define WALLT (-cos(angle*pi/180)) // virtual tension of facet on plane

constraint 1 /* the table top */
formula: x3 = 0
energy: // for contact angle
e1: -(WALLT*y)
e2: 0
e3: 0


constraint 2 NONNEGATIVE GLOBAL
function: x3=0


quantity facet_area info_only method edge_vector_integral global
vector_integrand:
q1: 0
q2: x
q3: 0


vertices
1 -0.4 -1.0 0.0 constraint 1 /* 8 vertices on plane */
2 0.4 -1.0 0.0 constraint 1
3 1.0 -0.4 0.0 constraint 1
4 1.0 0.4 0.0 constraint 1
5 0.4 1.0 0.0 constraint 1
6 -0.4 1.0 0.0 constraint 1
7 -1.0 0.4 0.0 constraint 1
8 -1.0 -0.4 0.0 constraint 1
9 -0.2 -0.5 0.5
10 0.2 -0.5 0.5
11 0.5 -0.2 0.5
12 0.5 0.2 0.5
13 0.2 0.5 0.5
14 -0.2 0.5 0.5
15 -0.5 0.2 0.5
16 -0.5 -0.2 0.5
17 0.0 0.0 1.0


edges /* given by endpoints and attribute */
1 1 2 constraint 1 /* 8 edges on plane */
2 2 3 constraint 1
3 3 4 constraint 1
4 4 5 constraint 1
5 5 6 constraint 1
6 6 7 constraint 1
```

```
7 7 8 constraint 1
8 8 1 constraint 1
9 9 10
10 10 11
11 11 12
12 12 13
13 13 14
14 14 15
15 15 16
16 16 9
17 1 9
18 2 10
19 3 11
20 4 12
21 5 13
22 6 14
23 7 15
24 8 16
25 9 17
26 10 17
27 11 17
28 12 17
29 13 17
30 14 17
31 15 17
32 16 17


faces /* given by oriented edge loop */
1 1 18 -9 -17
2 2 19 -10 -18
3 3 20 -11 -19
4 4 21 -12 -20
5 5 22 -13 -21
6 6 23 -14 -22
7 7 24 -15 -23
8 8 17 -16 -24
9 9 26 -25
10 10 27 -26
11 11 28 -27
12 12 29 -28
13 13 30 -29
14 14 31 -30
15 15 32 -31
16 16 25 -32


bodies /* one body, defined by its oriented faces */
1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 volume 1 density 1


read
```

```
re := {refine edges where on_constraint 1 }

inoutp := {printf "Facet area \t pressure \t # vertices \t p*A \n" >> "p1G001vertex.txt"
}
outp :={printf "%f10 \t %f10 \t %f10 \t %f10 \n",sum (edges where on_constraint 1,facet_area),
body[1].pressure, sum(vertices,1), sum (edges where on_constraint 1,facet_area)*body[1].pressure
>> "p1G001vertex.txt"}

procedure edge_equal (real tt) {
ll:=0;
facet_average :=0;
foreach edge ee do if on_constraint 1 then
{ll:=ll+1; facet_average := facet_average+max(ee.facet,area)};
facet_average := facet_average/ll;
refine edges ee where on_constraint 1 && 1.25*max(ee.facet,area)>facet_average;
K 0.4; u; edgeweed tt; u;
}

procedure gogo (real tt){
{{g25; u; edgeweed tt; u; outp; edge_equal(tt)} 100}
}

gogo2 := {r; g10; JIGGLE; {g100; u;} 4; JIGGLE OFF; r}
gogo2b := {{g5; u; V; u} 10; {{g20; u; V; u;} 10; edge_equal(0.05);} 5}
gogo2c := {{{g20; u; V; u;} 10; edge_equal(0.02);} 5}

goo := {gogo2; gogo2b 5; r; gogo2c 5; U; r; {gogo(0.01)} 5; {gogo(0.005)} 5; {gogo(0.002)}
10; {gogo(0.001)} 10;}

set facet color grey; set edge color red where on_constraint 1;
G 20;
```