Formation and Evolution of Breath Figures

Philipps Universität Marburg

MASTERARBEIT

dem Fachbereich Physik
der Philipps-Universität Marburg
vorgelegt

von

Johannes P. Blaschke
aus
Lich

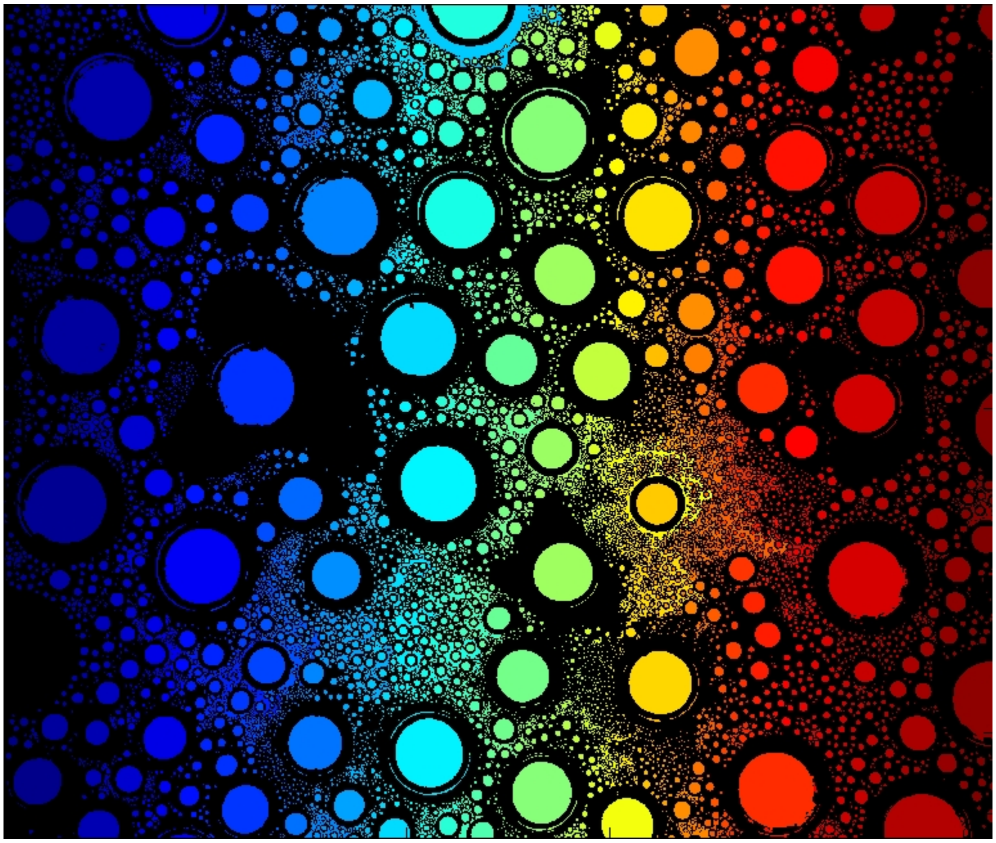
Marburg/Lahn 2010

Vom Fachbereich Physik der Philipps-Universität Marburg
als Masterarbeit angenommen am: 03.05.2010

Betreuer: Prof. Dr. Jürgen Vollmer
Zweitgutachter: Prof. Dr. Bruno Eckhardt

Formation and Evolution of

Breath Figures



"Problems worthy of attack
prove their worth by hitting back"

Piet Hein, "Problems," *Grooks* (1966)

Don't just read it! Ask your own questions, look for your own examples, discover your own proofs. Is the hypothesis necessary? Is the converse true? What happens in the classical special case? Where does the proof use the hypothesis?

Paul Halmos

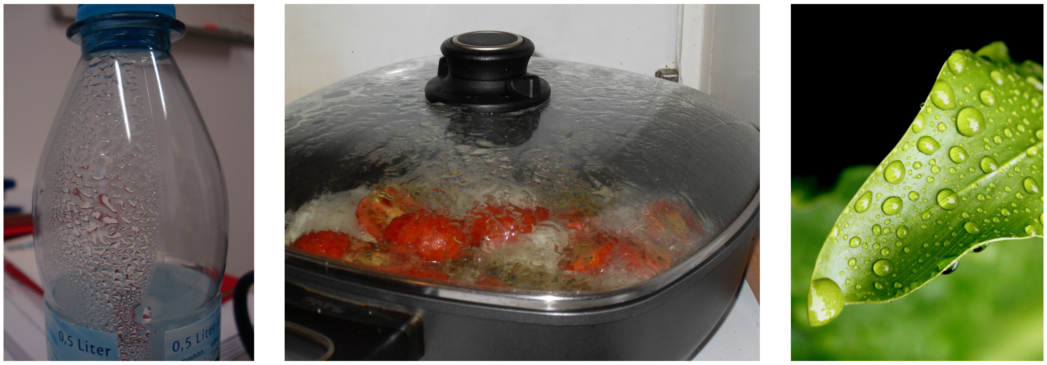
Contents

Chapter 1

Introduction

It matters not what goal you seek
Its secret here reposes:
You've got to dig from week to week
To get Results or Roses

Edgar Guest



(a) Condensation on the inner wall of a bottle of mineral water

(b) Condensation on the underside of a lid while cooking

(c) Dew on a leaf

FIGURE 1.1: Examples of breath figures in every day life.

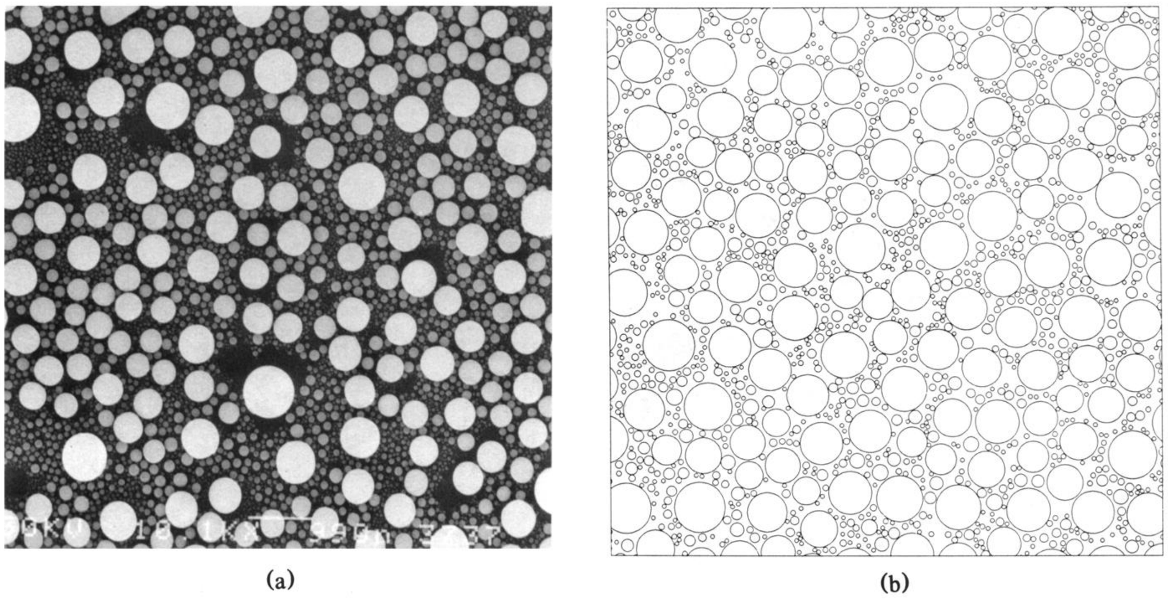


Figure 1.2: Visual agreement between size distribution of vapor-deposited tin on sapphire surface (a) with simulation (b). Taken from Family and Meakin [7]

1.1 Motivation and Background

The formation, growth and coarsening of "dew [2]" droplets on flat surfaces has widely been studied in the physics literature under the keyword "Breath Figures". Early work is in allusion to Rayleigh [13] and Aitken [1], who suggested the name "Breath Figures", due to an easy method of producing them in nature: the experimenter needs only to exhale on a relatively cool glass plate. Of course, there is no actual dependence on human *breath* as such, all that is necessary for breath figures to form is an over-saturated vapor being exposed to a substrate which provides conditions favorable to nucleation. Some particular every-day examples are shown in Fig. 1.1.

This makes breath figures almost everywhere vapor condenses, and thus unavoidable in nature. This also lends particular practical importance to breath figures wherever technological applications rely on the presence (or lack thereof) of condensation and depositions.

One of the earliest notable theoretical advances towards a theoretical description of breath figures was made by Family and Meakin [8], then later followed by Galvin and Briscoe [4] and [5]. A thorough discussion in the framework of renormalization groups has been given by Family and Meakin in the late 1980s. In order to facilitate analytical and numerical treatment they considered the following idealized setting:

1. there is a spatially uniform mass flux onto the surface

2. there is no evaporation or other removal of material from the surface
3. all droplets are spherical caps with a fixed wetting angle
4. the only interaction between droplets is collision and merging, which subsequently frees certain portions of the surface

Their results are summarized in the following subsection.

1.2 Scaling Theory by Family and Meakin

The Family-Meakin scaling theory suggests that the number of droplets of size[1] s at time τ is distributed as:

$$n(s,\tau) = s^{-\theta} f\left(\frac{s}{S(\tau)}\right) \tag{1.2.1a}$$

$$S(\tau) \sim \tau^{z} \tag{1.2.1b}$$

where $S(\tau)$ may be thought of as reflecting the largest droplet size encountered in the system. Its increase with time reflects the mass flux into the system. After all the theory assumes that no matter is allowed to leave the substrate once it has been deposited.

One may then estimate the scaling exponent θ from knowledge about the dimensionality of the substrate, d, and the droplets, D, such that the apparent radius of droplets is proportional to $R \sim s^{1/D}$ and the covered area on the substrate is $A \sim R^d$.

The theory assumes the conservation of 2 quantities: a) The wetted area is constant with time b) the total amount of mass in the system is proportional to the age of the system. The latter conserved quantity is simply a formal re-hash of the fact that there is a constant uniform flux, Φ, onto the surface and no material leaves the substrate. The former follows from consistency: the self-similarity ansatz Eq. (1.2.1a) describes a system for which a passage of time is equivalent to rescaling the length scale. And thus the area of area covered by the droplets needs to remain constant. Otherwise we would observe a different configuration of droplets (not accessible to rescaling) as time passes.

These conservation laws may be written formally:

$$\Phi\tau = \int s n(s,\tau)\,\mathrm{d}s \tag{1.2.2a}$$

$$W = \int s^{d/D} n(s,\tau)\,\mathrm{d}s \tag{1.2.2b}$$

[1] As a general note regarding notation: in the spirit of the work by Family and Meakin, the volume of a droplet is denoted the by s (read: "size"). To prevent confusion, when referring to the "size", s, of a droplet we will henceforth refer the actual *amount of material* or *mass* contained within a droplet.

where Φ is the flux of mass onto the substrate and W is the wetted area.

In order for the conserved quantities, Eqs. (1.2.2), to remain consistent with the scaling behavior in ansatz (1.2.1) we arrive at Theorem 1.1

Theorem 1.1. *Given a system with d-dimensional substrate and D-dimensional droplets, without droplets yielding, and with conserved quantities, Eqs. (1.2.2), subject to the scaling behavior in Eqs. (1.2.1) then:*

$$z(2-\theta) = 1 \tag{1.2.3a}$$

$$z(1+d/D-\theta) = 0 \tag{1.2.3b}$$

Proof. We start from the conserved quantity Eq. (1.2.2a) and apply our scaling ansatz:

$$\Phi\tau = \int sn(s,\tau)\,\mathrm{d}s = \int s^{1-\theta} f\left(\frac{s}{S(\tau)}\right)\,\mathrm{d}s \tag{1.2.4}$$

and apply the coordinate transformation: $x \equiv s/S(\tau)$ to change the variable of integration:

$$\begin{aligned}\int s^{1-\theta} f\left(\frac{s}{S(\tau)}\right)\,\mathrm{d}s &= \int (S(\tau)x)^{1-\theta} f(x)\, S(\tau)\,\mathrm{d}x \\ &= (S(\tau))^{2-\theta}\int x^{1-\theta} f(x)\;\mathrm{d}x \qquad (1.2.5)\end{aligned}$$

where the integral in Eq. (1.2.5) is considered to be time independent. Hence we have:

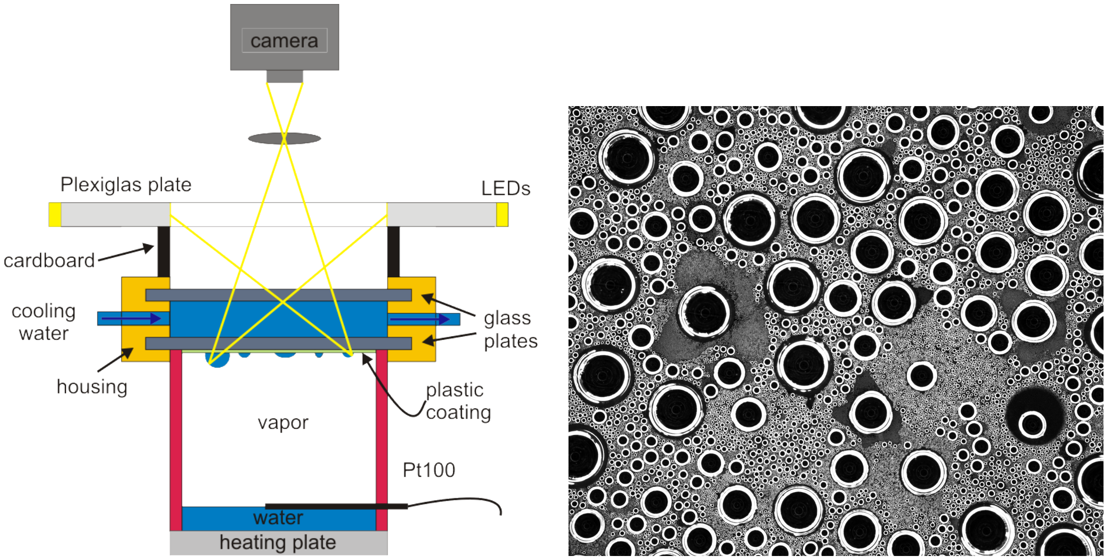
$$\tau \sim (S(\tau))^{2-\theta} \tag{1.2.6}$$

And considering $S(\tau) \sim \tau^z$ the result Eq. (1.2.3a) follows.

The result Eq. (1.2.3b) follows using the same reasoning, only stating the conserved quantity Eq. (1.2.2b). □

We may then apply Theorem 1.1 to our model system where $d = 2$ and $D = 3$ to find theoretical predictions for the scaling exponents: $\theta_{th} = \frac{5}{3}$ and $z_{th} = 3$. Family and Meakin's simulations find quantities close to this prediction: $\theta_{sim} = 1.56 \simeq \frac{5}{3} - 0.31$ and $z_{sim} = 2.92 = 3 - 0.08$.

The exponents encountered in our simulations are *near* the results predicted by the scaling theory. However there is a significant deviation. This deviation and the curiosity of how the findings might change due to the influence of gravity motivated experimental work in Göttingen.



(a) Experimental setup

(b) Image of breath figure on lid

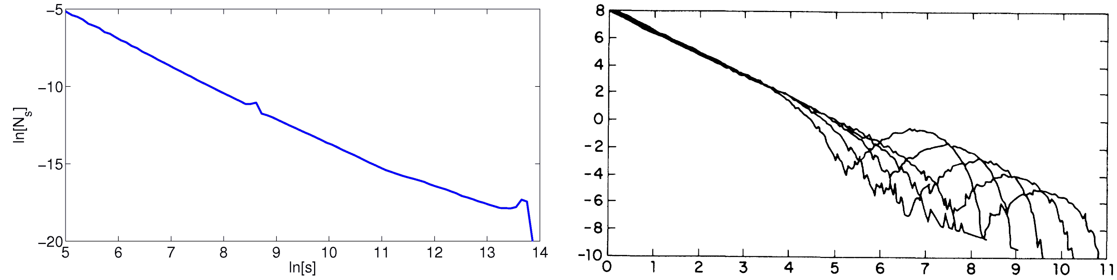
FIGURE 1.3: A schematic of the experimental setup of LAPP (a), together with a representative image, (b), of the droplets clinging to the underside of the cooled lid in (a).

1.3 Experimental Work in Göttingen

The research group in Göttingen has constructed a Rayleigh-Bénard cell with a transparent ceiling Fig. 1.3(a). It contains steam floating on a shallow layer of water, which is condensing on the cold lid of the cell. Droplets grow, merge and eventually drip down into the water basin when they reach a centimeter critical size. When running the setup for some time a steady state is reached, where one observes droplets of vastly different sizes Fig. 1.3(b). At regular instances images are taken with a video microscope, which are evaluated by image processing based on the Matlab Image Processing Toolbox. Presently, there are first data available for the size distributions of the water droplets forming on the ceiling Fig. 1.4(a). They are in a reasonable agreement with the shape of the droplet size distribution suggested by Family and Meakin Fig. 1.4(b).

However, *a priori* it is not clear whether Family and Meakin's results can apply to this experimental setup. After all, they considered a system which keeps coarsening in time, while the droplets in this system eventually drip off the cell's ceiling. Furthermore their models idealized the shape of, interaction between and flux onto droplets in order to facilitate the renormalization group treatment and to be able to numerically treat the model with the limitations of computing resources available twenty year ago. This motives us to strive for a more complete model incorporating more physical detail.

Furthermore, FAMILY AND MEAKIN's theory did not account for droplet sizes to have



(a) Measured size distribution based on pictures as shown in Fig. 1.3(b). The quantity $s = r^3$ is the size of the droplets in terms of perceived volume for a measured radius r.

(b) Simulated size distribution from Family and Meakin[8]. The quantity s is the actual volume of the droplets. The different lines refer to different times. The "bump" moves to the right with increasing time.

Figure 1.4: Comparison between the measurements taken by Lapp and that the simulations conduced by Family and Meakin[8].

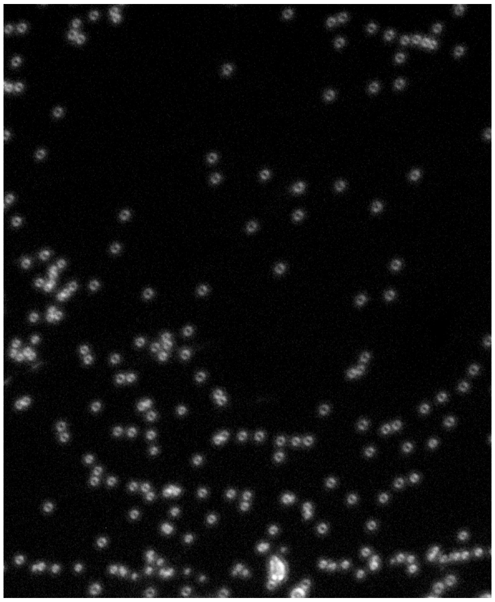


Figure 1.5: Freshly nucleated hanging droplets on a planar substrate. Since no smaller droplets can be seen in the voids, this appears to indicate a minimal size for which droplets can nucleate.

a lower bound at droplets of a *fixed* minimum size. Essentially droplets below this size cannot nucleate on a substrate, which is a direct contradiction to the scaling assumption underlying the proof of Theorem 1.1.

Lapp found that there appears to be a minimum size for droplets nucleating on a planar substrate. An example of this is demonstrated in Fig. 1.5, showing a minimum size for droplets nucleating in a patch of the substrate that has recently been cleared by coalescence or droplets falling off.

1.4 Aims of the Research Project

In our research project we will revisit the modeling of breath figures in order to address in more detail the physical processes behind the formation of droplets suspending from a flat level surface. We are looking for the most elementary generalization of the previous studies, which will allow us to address:

1. the theory of Family and Meakin, in order to gain a deeper understanding of the scaling behavior of breath figures when there is a lower cutoff to the droplet size, and understand why the simulated distributions deviate from the theoretical prediction.

2. how the droplet size distribution looks like in a steady state reached when particles attach to the lower side of a horizontal surface such that very large droplets are removed from the surface by gravity

3. the relevance of non-spherical (but still axially symmetric) droplet shapes arising due to gravitational forces acting on the big droplets in the system.

Chapter 2

Droplet Shapes

> Nothing at all takes place in the universe in which some rule of maximum or minimum does not appear
>
> Leonhard Euler

This work differentiates itself from previous works (such as Family & Meakin [8] and Beysens & Knobler [3]) in the sense that we explicitly address the effect of gravity — in particular for the formation of dew on the *underside* of a substrate. Hence it seems only natural to first ask *what sets hanging droplets apart from sessile ones in the presence of gravity?*

In essence, a system of hanging droplets exhibits:

1. a length scale beyond which droplet shape can no longer be approximated by a spherical cap, and
2. a fixed size characterizing the "yielding size"; the size of the largest droplet before the droplet mass would cause it to be torn from the substrate.

In this section we formulate a means of characterizing the shape of hanging droplets with a fixed contact angle and derive from this useful quantities such as:

1. the relationship between the observed radius and volume,
2. the maximal size of a droplet at a fixed contact angle (”Yielding Size”, or ”Yielding Radius”), and
3. the contact line forces.

2.1 Total Energy Functional Approach

When a droplet of some liquid is subject to a body force, in the static case it will assume a stationary state where the sum of all body forces are at balance with the surface tension. This can be most readily written down as minimizing the total energy as a functional of the enclosing surface.

2.1.1 Axially Symmetric Total Energy Functional

For an axially symmetric droplet, and in the absence of an interaction with a substrate (i.e., negligible contrast of liquid-substrate and gas-substrate surface tension), the total energy functional has a very intuitive and simple form, depending on which way we write the function describing the surface $\phi(x)$. Here we may consider two alternative approaches summarized in Fig. 2.1(a) and Fig. 2.1(b):

a) The free parameter is the radial distance, thus r is the radial distance and $\phi(r)$ is the "depth" of the droplet at a distance r from the center (Fig. 2.1(a)).

b) The free parameter is the axial distance, thus h is the axial distance and $\phi(h)$ is the "breadth" of the droplet at a distance h from the substrate (Fig. 2.1(b)).

In either case, we may write the total energy functional independently of the chosen coordinate system as follows:

Theorem 2.1. *Let $\gamma_{\mu\nu}$ be the metric describing the surface of a droplet, let ω be the domain of a parametrization of γ and let Ω be the region bounded by γ. Then the total energy functional for such a droplet attached to a substrate and subject to a body force of magnitude g acting in direction x_3 and surface tension σ between the droplet and surrounding gas phase is given by:*

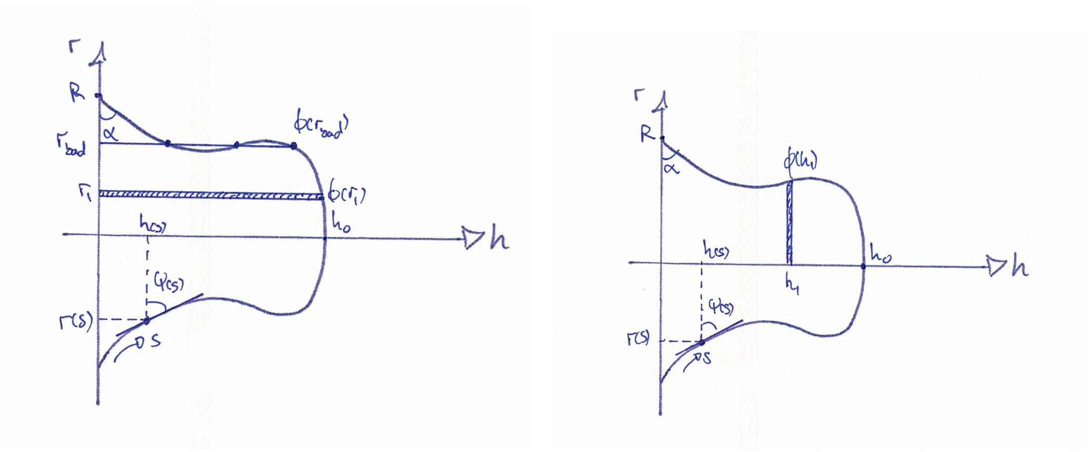
$$F[\gamma] = \int\int_{\omega} \left(\sigma\sqrt{\det\gamma}\right)\ \mathrm{d}x_1\ \mathrm{d}x_2 - \int\int\int_{\Omega} \rho g x_3\ \mathrm{d}x_1\ \mathrm{d}x_2\ \mathrm{d}x_3 \tag{2.1.1}$$

Proof. The total energy of a droplet, bounded by a surface $\partial\Omega$ and subject to a body force described by a potential U, is:

$$F = \sigma|A| + \int_{\Omega} U\ \mathrm{d}\Omega \tag{2.1.2}$$

where $|A|$ is the total surface area of the droplet. It is then a simple application of calculus in curvy-linear coordinates to show:

$$|A| = \int\int_{\omega} \left(\sigma\sqrt{\det\gamma}\right)\ \mathrm{d}x_1\ \mathrm{d}x_2 \tag{2.1.3}$$



(a) An axially-symmetric droplet profile characterized by a function as described by case 2.1.1. The shaded slither will have a total gravitational potential energy given by: $-\int_0^{\phi(r_1)}\int_0^{2\pi} \rho g h r \, \mathrm{d}\theta \, \mathrm{d}h$. The fatal case described later is made apparent by the multi-valued nature of $\phi(r_{bad})$.

(b) An axially-symmetric droplet profile characterized by a function as described by case 2.1.1. The shaded slither will have a total gravitational potential energy given by: $-\int_0^{\phi(h_1)}\int_0^{2\pi} \rho g h r \, \mathrm{d}\theta \, \mathrm{d}r$. What was a fatal case in 2.1(a) is no longer present here as the function $\phi(h)$ remains single valued for all values of h.

FIGURE 2.1: Parametrization of surface integrals, and the parametrization used in WENTE theory. Note that in this coordinate system, gravity is directed towards the left. Hence the integrals potential energy having a negative value.

where γ is the metric of the surface and ω is the domain of its parametrization. Furthermore, if we restrict the body force along the direction $\mathrm{d}x_3$, our potential U may be written as $\rho g x_3$ for a liquid mass with density ρ in a gravitational field g. □

Eq. (2.1.1) is nothing more than the expression that energy can be gained by going deeply into the gravitational potential, and that energy is lost for every unit surface area that the droplet possesses. Interactions with the substrate have not been included other than the substrate providing a boundary of integration. In other words, the substrate provides us with an *axis* bounding our integral. This is not a physically correct method for dealing with the interaction energy between droplet and substrate as energy can be gained/lost when the liquid is wetting the substrate (depending on whether our substrate is hydrophilic or hydrophobic). In section 2.2.1 we will show, how to augment the approach to account for these contributions.

The parametrization shown in Fig. 2.1(a) has a very intuitively appealing form and so we shall examine it first. For this case, the metric $\gamma_{\mu\nu}$ has the following properties:

$$\begin{aligned} \gamma_{hh} &= \phi'^2(r) \\ \gamma_{rr} &= 1 \end{aligned} \tag{2.1.4}$$

and 0 elsewhere, where the r and h directions are defined in Fig. 2.1(a). Also choosing cylindrical coordinates centered about the axis h we may write the Jacobian determinant as: $\det J = r$.

Hence we can write the total energy functional as:

$$\begin{aligned} F[\phi(r)] &= \int_0^R \int_0^{2\pi} \left(\sigma\sqrt{1+\phi'^2}\right) r \, \mathrm{d}\theta \, \mathrm{d}r - \int_0^R \int_0^{\phi(r)} \int_0^{2\pi} \rho g h r \, \mathrm{d}\theta \, \mathrm{d}h \, \mathrm{d}r \\ &= \pi \int_0^R r \left(2\sigma\sqrt{1+\phi'^2} - \rho g \phi^2\right) \, \mathrm{d}r \end{aligned} \tag{2.1.5}$$

where $\phi(r)$ is the depth the droplet at a radial distance r from its center, σ is the surface tension of the liquid phase, ρ is the density of the liquid phase, R is the radius of the wetted region of the substrate, and g is gravitational constant.

While the formulation based on the parametrization in Fig. 2.1(a) is probably the most intuitive, it is not the most universal as undulations would make $\phi(r)$ no longer a *function* (example shown in Fig. 2.1(a)). Since undulations are commonly seen for these surfaces [10], it would seem sensible to formulate our problem in such a light that $\phi(x)$ remains a function for all parameters that we wish to choose. This motivates us to formulate the total energy functional as shown in Fig. 2.1(b).

In the parametrization detailed in Fig. 2.1(b), the metric $\gamma_{\mu\nu}$ has the following properties:

$$\begin{aligned} \gamma_{rr} &= \phi'^2(h) \\ \gamma_{hh} &= 1 \end{aligned} \tag{2.1.6}$$

and 0 elsewhere. Where the r and h directions are defined in Fig. 2.1(b). Also choosing cylindrical coordinates centered about the axis h we may write the Jacobian determinant as: $\det J = \phi(h)$.

Hence, in such coordinates, Eq. (2.1.1) would become:

$$\begin{aligned} F[\phi(h)] &= \int_0^{h_0} \int_0^{2\pi} \left(\sigma\sqrt{1+\phi'^2}\right) \phi(h) \, \mathrm{d}\theta \, \mathrm{d}h - \int_0^{h_0} \int_0^{\phi(h)} \int_0^{2\pi} \rho g h r \, \mathrm{d}\theta \, \mathrm{d}r \, \mathrm{d}h \\ &= \pi \int_0^{h_0} \phi(h) \left(2\sigma\sqrt{1+\phi'^2} - \rho g h \phi(h)\right) \, \mathrm{d}h \end{aligned} \tag{2.1.7}$$

As the static shape of a droplet is given by the surface which minimizes the total energy functional of said droplet, in order to determine the shape of the surface of such a droplet, all that now needs to be done is to determine the surface such that the total energy functional is stationary:

$$\frac{\delta F[\phi(x)]}{\delta\phi(x)} \equiv 0 \tag{2.1.8}$$

According to the Euler-Lagrange formalism, condition (2.1.8) is satisfied for the total energy functional Eq. (2.1.7) iff:

$$2\sigma\sqrt{1+\phi'^2(h)} - \rho g h\phi(h) - \frac{\mathrm{d}}{\mathrm{d}h}\left(\frac{2\sigma\phi(h)\phi'(h)}{\sqrt{1+\phi'^2(h)}}\right) = 0 \tag{2.1.9}$$

2.2 Wente's Parametrization

An approach to solve the differential equation (2.1.9) has been suggested by Wente [17] who has proven the following theorem:

Theorem 2.2. *Let the parameter $s(\tau) \equiv \int_0^\tau \sqrt{1+\phi(x)'^2}\,\mathrm{d}x$ be the distance along the curve $\phi(x)$ starting the integral from the point $(0, h_0)$ and integrating towards the substrate. Let ψ be the angle between the tangent to $\phi(x)$ and the substrate, and let h be the axial coordinate and r be the radial coordinate of the parametric curve for $\psi(x)$ (as defined in Fig. 2.1(a) and Fig. 2.1(b)). Then $\phi(x)$ is a surface satisfying (2.1.8) for (2.1.5) or (2.1.7) iff:*

$$\frac{\mathrm{d}\psi}{\mathrm{d}s} = -h - \frac{\sin\psi}{r} \tag{2.2.10a}$$

$$\frac{\mathrm{d}r}{\mathrm{d}s} = \cos\psi \tag{2.2.10b}$$

$$\frac{\mathrm{d}h}{\mathrm{d}s} = \sin\psi \tag{2.2.10c}$$

Where r and h are measured in units of capillary length.

Proof. We may simplify the rather nasty Eq. (2.1.9) using the parametrization described in Theorem 2.2 and Fig. 2.1(b):

As Eq. (2.1.9) holds at the surface of our droplet, it is more sensible to describe the parametrized surface of the droplet $(h(s), r(s))$ as solving a set of differential equations defined on $(h(s), r(s))$. Firstly, as ψ is the angle of the surface to the substrate, we find:

$$\frac{1}{\phi'(h)} = \frac{\mathrm{d}h}{\mathrm{d}r} = \tan\psi \quad\Rightarrow\quad \frac{\mathrm{d}s}{\mathrm{d}h} = \sqrt{1+\phi'^2(h)} = \sqrt{1+\frac{1}{\tan^2\psi}} \tag{2.2.11}$$

and using this, we may now re-express Eq. (2.1.9) in terms of our parametrized curve

$(h(s), r(s))$:

$$\begin{aligned}\frac{\mathrm{d}s}{\mathrm{d}h} &= \sqrt{1+\frac{1}{\tan^2\psi}} \\ &\Rightarrow \frac{\mathrm{d}h}{\mathrm{d}s} = \sin\psi \qquad (2.2.12) \\ \frac{\mathrm{d}r}{\mathrm{d}s} &= \frac{\mathrm{d}r}{\mathrm{d}h}\frac{\mathrm{d}h}{\mathrm{d}s} = \frac{\mathrm{d}r}{\mathrm{d}h}\sin\psi = \frac{1}{\tan\psi}\sin\psi \\ &\Rightarrow \frac{\mathrm{d}r}{\mathrm{d}s} = \cos\psi \qquad (2.2.13)\end{aligned}$$

whereas the equation for ψ is somewhat more inspired. Starting from Eq. (2.1.9), we may apply the parametrization discussed in Fig. 2.1, namely $(\phi(x), x) = (r(s), h(s))$ This means that $\phi\left(h(s)\right) = r(s)$:

$$\begin{aligned}2\sigma\sqrt{1+\phi'^2(h)} - \rho g h\phi(h) - \frac{\mathrm{d}}{\mathrm{d}h}\left(\frac{2\sigma\phi(h)\phi'(h)}{\sqrt{1+\phi'^2(h)}}\right) &= 0 \\ \Rightarrow \sqrt{1+r'^2} - \frac{\mathrm{d}}{\mathrm{d}h}\left(\frac{r\frac{\mathrm{d}r}{\mathrm{d}h}}{\sqrt{1+r'^2}}\right) &= -\frac{\rho g}{\sigma}hr \qquad (2.2.14)\end{aligned}$$

Re-expressing all units in terms of the capillary length $\lambda_c = \sqrt{\frac{\sigma}{\rho g}}$ yields:

$$\begin{aligned}\frac{1}{\sin\psi} - \frac{\mathrm{d}}{\mathrm{d}h}\left(\frac{r\frac{1}{\tan\psi}}{\frac{1}{\sin\psi}}\right) &= -hr \\ \Rightarrow \frac{1}{\sin\psi} - \frac{\mathrm{d}}{\mathrm{d}h}\left(r\cos\psi\right) &= -hr \\ \Rightarrow \frac{1}{\sin\psi} - \frac{\mathrm{d}r}{\mathrm{d}h}\cos\psi + r\frac{\mathrm{d}}{\mathrm{d}h}\left(\cos\psi\right) &= -hr \\ \Rightarrow \frac{1}{\sin\psi} - \frac{\cos\psi}{\tan\psi} + r\sin\psi\frac{\mathrm{d}\psi}{\mathrm{d}h} &= -hr \\ \Rightarrow \frac{\mathrm{d}\psi}{\mathrm{d}h} = -h - \frac{\sin\psi}{r} \qquad (2.2.15a)\end{aligned}$$

Where the last step relies on: $\frac{1}{\sin\psi} - \frac{\cos\psi}{\tan\psi} = \sin\psi$ and Eq. (2.2.12). Thus completing the proof[1]. □

Also, nothing has been said about the boundary values of this problem. We will use the convention that $s(\tau = 0) = 0$ represents the lowest point on the droplet, and that $s(\tau = h_0) = S$ represents the point where the droplet meets the substrate such that $h(S) = 0$.

[1] We remark: the same approach can be applied to Eq. (2.1.5), yielding the same result with somewhat simpler algebra. However, as the parametrization shown in Fig. 2.1(a) is only well defined for a subset of shape profiles the proof will be omitted here.

In other words, physically realistic solutions are ones that solve Eq. (2.2.10a) to Eq. (2.2.10c) under the constraint that:

$$\psi(0) = 0 \tag{2.2.16a}$$
$$r(0) = 0 \tag{2.2.16b}$$
$$h(0) = h_0 \tag{2.2.16c}$$

and upon integration[2] we find:

$$\psi(S) = \alpha \tag{2.2.18a}$$
$$r(S) = R \tag{2.2.18b}$$
$$h(S) = 0 \tag{2.2.18c}$$

where α is the contact angle with the substrate.

When computing the shape profiles defined by Eq. (2.2.10) for different h_0 values and subject to constraints (2.2.16), one notices a spread of contact angles at the substrate. This observation is actually sensible as Eq. (2.1.1) has ignored the interaction between the substrate and the droplet which fixes the contact angle. This motivates us to extend the method to find droplet shapes with a prescribed contact angle.

2.2.1 Including Contact Angle in Wente's Solution

Modifying Eq. (2.1.1) to include a term for the energy due to an interaction between droplet and substrate would require numerical solutions to be somewhat more laborious, as they usually require self-consistent solutions to be found for some discretized surface elements. The elegance of solutions discussed in section 2.1.1 lies in the fact that they may be integrated with relative ease, using programs such as MATLAB© leaving the physicist time for physics rather than programming.

An inspiration into how contact angles may be included in the method by Wente comes from realizing that contracting the r axis and stretching the h axis would in

[2] There is a subtlety to be taken into consideration when computing solutions numerically: At the point $(r, h) = (0, h_0)$, Eq. (2.2.10a) has a singularity which fortunately is a removable singularity:

$$\lim_{s\to 0} \frac{\mathrm{d}\psi}{\mathrm{d}s} = 0 \tag{2.2.17}$$

i.e., the profile becomes flat at the lowest point, and therefore does not pose a formal concern to the regularity of this problem. However, numerical methods such as Runge-Kutta methods can quickly become unstable near the point $(r, h) = (0, h_0)$ and yield divergent solutions due to numerical errors. Hence care needs to be taken at the point $(r, h) = (0, h_0)$. Ergo, it is a wise strategy to require Eq. (2.2.16a) to read $\psi(\epsilon) = 0$ (allowed because of 2.2.17) and integrate from there until such a τ such that $h(s(\tau)) = 0$.

fact produce a different contact angle in these new coordinates. Of course one needs to know *how* equations (2.2.10a), (2.2.10b), and (2.2.10c) change under said coordinate transformation.

In this section a primed-derivative, $\phi'(h)$ denotes the derivative with respect to untransformed coordinates, whereas a dotted-derivative, $\dot{\phi}(H)$, denotes a derivative with respect to a transformed coordinate.

Definition 2.1. *For some dilation factor, $q \in \mathbb{R}$, let capital letters*[3] *denote the coordinates obtained by transforming their lower-case counterparts introduced in Fig. 2.1(a) and Fig. 2.1(b) subject to the transformation:*

$$R \equiv q \cdot r \tag{2.2.19a}$$

$$H \equiv \frac{1}{q} \cdot h \tag{2.2.19b}$$

$$S(\tau) \equiv \int_0^\tau \sqrt{1 + \dot{\phi}^2(X)}\, \mathrm{d}X \tag{2.2.19c}$$

where Eq. (2.2.19c) refers to the distance along our droplet surface in the new coordinates, the integral is performed in the new coordinates.

Lemma 2.1. *Given the coordinate transformation definition 2.1, derivatives transform as follows:*

$$\frac{\mathrm{d}}{\mathrm{d}r} = q\frac{\mathrm{d}}{\mathrm{d}R} \tag{2.2.20a}$$

$$\frac{\mathrm{d}}{\mathrm{d}h} = \frac{1}{q}\frac{\mathrm{d}}{\mathrm{d}H} \tag{2.2.20b}$$

Proof. Application of the chain rule. □

Lemma 2.2. *In the spirit of the convention outlined in definition 2.1 we interpret Ψ as the angle between the substrate and the tangent of the droplet surface measured in the transformed coordinates giving:*

$$\frac{1}{\dot{\phi}(H)} = \frac{\mathrm{d}H}{\mathrm{d}R} = \tan\Psi \tag{2.2.21}$$

Proof. Repeated application of lemma 2.1 on the parametrization used in the proof for theorem 2.2. □

With these lemmas we now have an understanding that the quantities in the transformed coordinates can be parametrized in a similar manner as with the surface in the untransformed coordinates. The intuitive step here, is that we may *adjust* the contact angle of

[3] Note, the symbol R also refers to the wetting radius of our droplets (refer to Fig. 2.1). This should not lead to any confusion, for the convention that capital letters refer to transformed coordinates is only used in this subsection.

an already determined surface by squeezing it lengthwise and stretching it breadthwise. All that now remains to be done is determine how Equations (2.2.10a), (2.2.10b), and (2.2.10c) change under the transformation, this is done with the following theorem:

Theorem 2.3. *Let $q \in \mathbb{R}$, let the parameter $S(\tau) \equiv \int_0^\tau \sqrt{1+\dot{\phi}^2(X)}\,\mathrm{d}X$ be the distance along the curve $\phi(X)$, let Ψ be the angle between the tangent to ϕ and the substrate, and let H be the axial coordinate and R be the radial coordinate of the parametric curve for $\psi(X)$ [analogue to the lower-case units defined in Fig. 2.1(a) and Fig. 2.1(b)]. Then $\phi(X)$ is a surface satisfying (2.1.8) for (2.1.5) or (2.1.7) iff:*

$$\frac{\mathrm{d}\Psi}{\mathrm{d}S} = -\left(\frac{\lambda^3(\Psi,q)}{q^2}\right)H - \left(\lambda^2(\Psi,q)\right)\frac{\sin\Psi}{R} \tag{2.2.22a}$$

$$\frac{\mathrm{d}R}{\mathrm{d}S} = \cos\Psi \tag{2.2.22b}$$

$$\frac{\mathrm{d}H}{\mathrm{d}S} = \sin\Psi \tag{2.2.22c}$$

$$\lambda(\Psi,q) = \sqrt{\cos^2\Psi + q^4\sin^2\Psi} \tag{2.2.22d}$$

where R and H are measured in units of capillary length.

Proof. By application of Lemmas 2.1 and 2.2 to Eq. (2.2.14), we get the Euler-Lagrange equation for our system in the transformed coordinates:

$$-\frac{1}{q^2}\sqrt{q^4+\dot{R}^2} - \frac{1}{q^2}\frac{\mathrm{d}}{\mathrm{d}H}\left(\frac{R\frac{\mathrm{d}R}{\mathrm{d}H}}{\sqrt{q^4+\dot{R}^2}}\right) = -HR \tag{2.2.23}$$

Using the definition of the angle in the new coordinates, $\dot{H} = \tan\Psi$, and the natural parametrization of the curve in the new coordinates $\frac{\mathrm{d}S}{\mathrm{d}R} = \sqrt{1+\dot{H}^2}$, the same algebra utilized in the proof for Theorem 2.2 yields that, given Eq. (2.2.23):

$$\frac{\mathrm{d}R}{\mathrm{d}S} = \cos\Psi \tag{2.2.24}$$

and:

$$\frac{\mathrm{d}H}{\mathrm{d}S} = \sin\Psi \tag{2.2.25}$$

It is then a matter of more tedious but still trivial algebra to find[4]:

$$\frac{\mathrm{d}\Psi}{\mathrm{d}S} = -\left(\frac{\lambda^3(\Psi,q)}{q^2}\right)H - \left(\lambda^2(\Psi,q)\right)\frac{\sin\Psi}{R} \tag{2.2.26}$$

completing the proof. □

[4] Following essentially the same tricks used in the derivation (2.2.15) and recognizing the form of Eq. (2.2.22d)

The ODEs found by WENTE are unit-free, hence each solution to WENTE's equations (2.2.10) represents a class of droplets where the capillary length would be used to relate this unit-free solution to real-world materials. Hence, how theorem 2.3 ought to be interpreted is that the parameter q makes the radial and axial directions have slightly different lengths, in turn changing the contact angle. In order to ensure that the same *physics* takes place in the new coordinates, the coupled ODEs which determine the surface are also changed. Yet they remarkably retain most of their form except for a shape distortion function $\lambda(\Psi, q)$.

The droplet volume is not conserved by this transformation, in spite of this, this does not constitute a problem since quantities such as radius of the wetted area and volume are quantities that are "measured" from the droplet profiles. The user of this model only fixes the depth of the droplets and their contact angle calculating the droplet profile, from this they derive useful quantities such as radius and volume which can then be plotted against each other. In this sense, the droplet depth enumerates the set of all droplet profiles which can then be analyzed for the quantities of interest.

2.3 Shape of Hanging Droplets

Now we have a means to using WENTE's solution on real-world problems which actually interact with the substrate to which they cling. This interaction is phenomenologically modeled using the contact angle α, which is a quantity depending on the materials of the liquid phase and the substrate, and the morphology of the substrate.

2.3.1 Applying the Modified Wente Equations

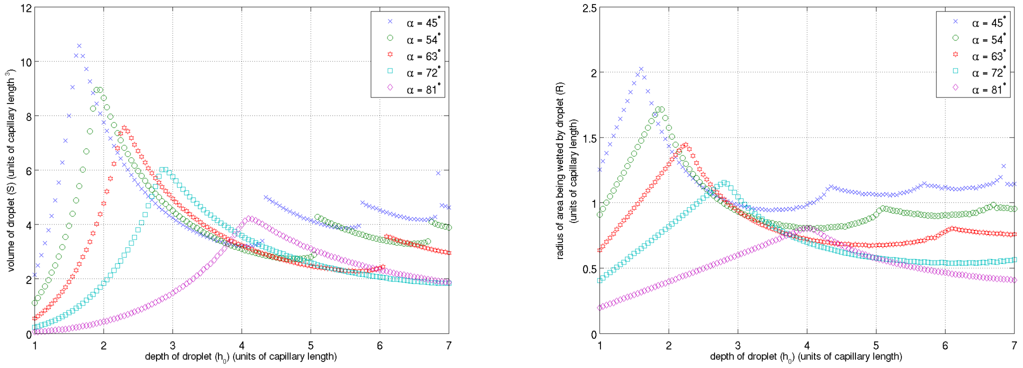
The factor q only implicitly determines the contact angle α. Hence the function $q(\alpha)$ is not known explicitly. It turns out, however, that for any contact angle α of interest it is relatively straight forward to guess solutions ϕ_{q_1} and ϕ_{q_2} with contact angles α_{q_1} and α_{q_2} respectively, such that $\alpha_{q_1} < \alpha < \alpha_{q_2}$. Consequently, bisection iteration lends itself as a suitable means for improving the estimate for the q yielding our desired α[5].

This provides us with a simple algorithm for finding solutions with fixed contact angle:

1. Let α be our desired contact angle. Determine an upper bound, q_u, and a lower bound, q_l, solutions for the q factor.

2. Determine the contact angle for the upper and lower bound solutions, with contact angles α_u and α_l respectively.

[5] An important limitation of using this method is that up to now, we could not find profiles with a angle greater than 90°. Consequently, profiles with contact angles greater than 90° are not accessible.



(a)

(b)

FIGURE 2.2: The volume (Fig. 2.2(a)) and the radius (Fig. 2.2(b)) of different solutions to WENTE's problem whilst holding contact angle fixed at the values shown. The different h_0 values represent different solutions (and also representing the depth of the droplets).

3. Iterate the following algorithm until $|\alpha - \alpha_m|$ is less than the error tolerance (typically 10^{-6} for this work).
   (a) Compute the mid-point solution for $q_m \equiv \frac{q_u+q_l}{2}$ and determine its contact angle α_m
   (b) If $\alpha_m - \alpha > 0$ then re-assign the mid-point solution to the upper-bound
   (c) else re-assign the mid-point solution to the lower bound.

2.3.2 Results

The model parameter which determines the solution found by integrating Eq. (2.2.22) is the starting point h_0. Hence $\{h_0\}$ enumerates the solutions for the WENTE problem. We can now apply the algorithm discussed in section 2.3.1 to fix the contact angle while varying h_0. Moreover we can extract useful measurable quantities such as volume and radius of the wetted area (henceforth this shall be referred to as just "radius") from these profiles. These shall be discussed here:

We may then combine the relationships $S(h_0)$ and $R(h_0)$, shown in the two panels of Fig. 2.2 to determine the quantity $S(R)$, the dependence of volume on radius, by interpreting h_0 as the parameter of a parametrized version of the curve of $S(R)$, simply by using: $(R, S(R)) = (R(h_0), S(h_0))$. This yields Fig. 2.3.

Fig. 2.3 surely shows a rather curious behavior: there are multiple solutions for the volume of a droplet for a given volume. The reader ought to think of going to these more exotic solutions by starting at the lower branch of Fig. 2.3 and then traveling

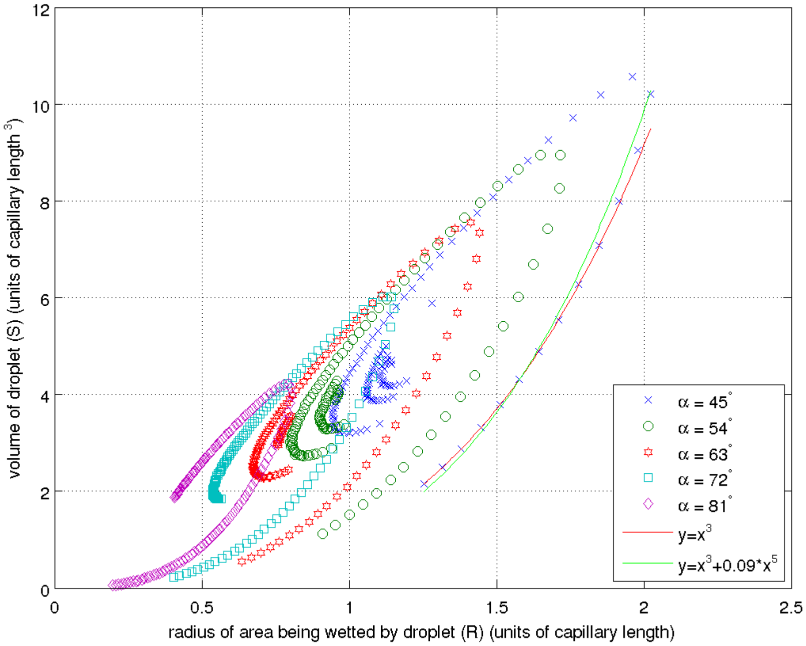
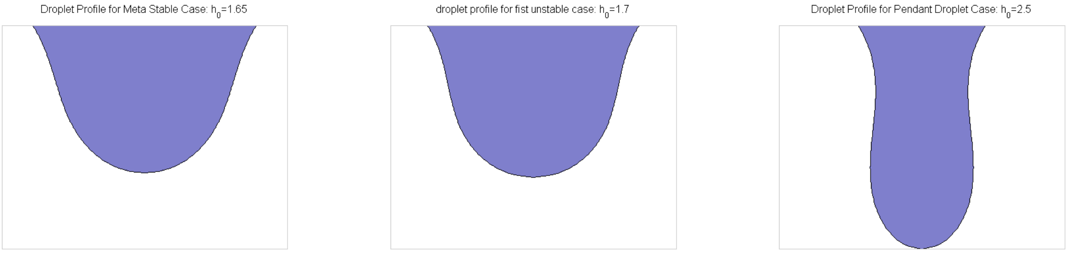


FIGURE 2.3: The volume dependence on radius of different solutions to WENTE's, $S(R)$, problem whilst holding contact angle, α, fixed at the values shown. The observed cycles that there are multiple solutions for given S. Bifurcations in the shape of the profiles have been observed whenever the cycle is interrupted. At these points the pendant droplet develops another "pendant". Solid lines represent both the relation that $S(R) \sim R^3$ (red line) and $S(R) \sim R^3 + R^5$ (green line) to aid the reader in observing the transition from spherical caps to more elongated profiles.

counter-clockwise to solutions of higher volume. As discussed later, these solutions are not stable against axial perturbations and therefore are not physically realizable. However this is still an interesting mathematical curiosity which can be understood by examining the shape of the droplet. Fig. 2.4 shows the shape of droplets for different h_0 parameters.

From the transition from Fig. 2.4(b) to Fig. 2.4(c) we can see a clear difference in shape, most notably: the development of inflection. This causes most of the volume to become displaced in the "pendant" rather than near the surface, and so the apparent contact radius is reducing with increasing volume. If one was to take this a step further, we could see a profile shown by Fig. 2.5 which represent the creation of new pendants and correspond to the saddle-node bifurcations seen in figures 2.2(a) to 2.3. See WENTE [17] for a more detailed discussion of these unstable solutions. However, as these profiles are far from stable, they are not physically relevant. We have shown these only for mathematical completeness.

We now return to the subject of the function $V(r)$ as shown in Fig. 2.3. One question

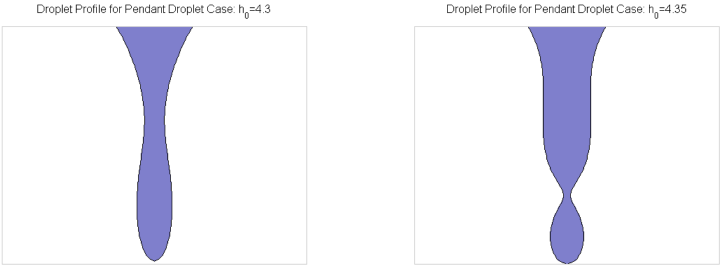


(a) Metastable droplet. In capillary length units: $h_0 = 1.65$, $R = 1.96$

(b) First unstable droplet. In capillary length units: $h_0 = 1.70$, $R = 1.85$

(c) Pendant droplet. In capillary length units: $h_0 = 2.50$, $R = 1.12$

FIGURE 2.4: Profiles of droplets for fixed contact angle $\alpha = 45°$: (a) metastable droplet just before it becomes unstable; (b) unstable case; (c) unstable case displaying a characteristic inflection, which is often called a "pendant" droplet. The change in radius from (a) to (b) indicates that the least stable droplet also has the largest wetting radius.



(a) Shortly before bifurcation. $h_0 = 4.30$(units of capillary length)

(b) Shortly after bifurcation. $h_0 = 4.35$ (units of capillary length)

FIGURE 2.5: Saddle-Node bifurcation causing the "jumps" visible in Fig. 2.3. These cases are plotted for a contact angle of $\alpha = 45°$. At critical depths h_0, the shape discontinuously changes, causing the formation of an additional point of inflection. This is responsible for discontinuities in functions such as the volume $S(R)$ or the energy of the droplets. Of course these droplet profiles are far from being stable.

that greatly concerns the experiments by LAPP is *at which radii do the droplets stop being spherical caps?* One method would be to include a high order-dependence on radius in the equation form volume:

$$S(R) \simeq c(\alpha) R^3 \left(1 + \frac{\kappa}{c(\alpha)} R^2\right) \tag{2.3.27}$$

Eq. (2.3.27) may be considered an approximation to the "true" function $V(r)$: We consider the total energy functional

$$\begin{aligned} F[\phi(h)] &= \pi \int_0^{h_0} \phi(h) \left(2\sigma\sqrt{1+\phi'^2} - \rho g h \phi(h)\right) \,\mathrm{d}h \\ F[\phi(h)]] &= 2\pi \int_0^{h_0} \phi(h)\sigma\sqrt{1+\phi'^2}\,\mathrm{d}h - \pi \int_0^{h_0} \rho g h \phi^2(h)\,\mathrm{d}h \end{aligned} \tag{2.3.28}$$

where the first integral goes as $O\left(\sigma R^2\right)$ and the second integral scales like $O\left(\rho g R^4\right)$, R being the wetting radius of our droplets. Hence the second term is a correction of order $\epsilon \equiv O\left(\frac{\rho g}{\sigma} R^2\right) = O\left(\frac{R^2}{\lambda_c^2}\right)$. Thus corrections to our volume are expected to take the form:

$$S(R) = c(\alpha)R^3\left(1 + \epsilon + O\left(\epsilon^2\right)\right) = c(\alpha)R^3\left(1 + O\left(\frac{R^2}{\lambda_c^2}\right) + O\left(R^4\right)\right) \tag{2.3.29}$$

hence indicating that the first correction ought to be of order 5 with respect to the wetted radius.

The term $c(\alpha)$ is a term depending on the contact angle and would be $\frac{2\pi}{3}$ for a contact angle of 90°. $c(\alpha)$ may be given analytically as demonstrated by Tadmor [14]:

$$c(\alpha) = \frac{\pi}{3}\left(\frac{2 - 3\cos\alpha + \cos^3\alpha}{\sin^3\alpha}\right) \tag{2.3.30}$$

As Fig. 2.6 shows, there is a transition at about 0.6 times the capillary length.

It is also clear form Fig. 2.6 that κ in Eq. (2.3.27) can be used to determine when the transition from spherical caps occurs. As the contact angle in many experiments is poorly defined (either due to contact angle hysteresis or because of contact line pinning), or difficult to measure as in the case of Lapp, it may be a suitable fitting parameter. Hence, we studied the dependence of κ on α as shown in Fig. 2.7.

2.4 Stability and Yielding of Hanging Droplets

Another quantity of great interest is the radius at which droplets pinch off due to surface tension forces no longer being able to withstand the droplet's weight (henceforth this shall be referred to as "yielding radius").

In order to find this quantity, one first needs to define a stability condition for hanging droplets. This is most readily determined from the total energy functional Eq. (2.1.1) using Eq. (2.4.31).

Lemma 2.3. *Given a function $\phi_{h_0\alpha}$ which minimizes Eq. (2.1.1) according to Eq. (2.1.8), and letting $E(h_0) = F[\phi_{h_0\alpha}]$ then $\phi_{h_0\alpha}$ is physically stable if:*

$$\left(\frac{\partial E(h_0)}{\partial h_0}\right)_{\alpha,\,S} > 0 \tag{2.4.31}$$

where the derivative is taken while holding the contact angle α and volume S constant.

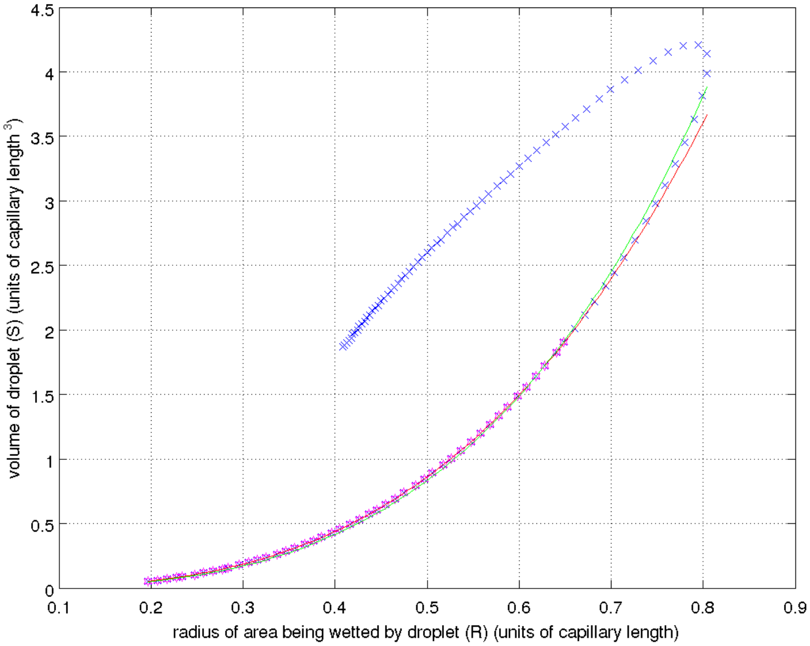


FIGURE 2.6: The actual function $S(R)$ for $\alpha = 81°$ showing the transition from spherical caps (red line) to a higher-order function (approximated by Eq. (2.3.27)). Magenta points have been used in fitting Eq. (2.3.27), which then extrapolates well for greater radii. For a radius of approx. 0.6 capillary lengths the droplet no-longer may be considered a spherical cap.

Proof. If Eq. (2.4.31) is negative, then allowing the lowest point of the droplet to move lower is energetically favorable. Thus, while such solutions represent stationary surfaces according to Eq. (2.1.8), these would not be stable against small perturbations in the axial direction. □

2.4.1 Numerical Data

We may now apply Eq. (2.4.31) to our numerical data, giving Fig. 2.8. This may be interpreted using Theorem 2.3.

Our data shows a clear cutoff after which solutions are no longer physically stable as they would loose potential energy by moving their lowest point deeper.[6]

[6] Yet, there are h_0 parameters for which the droplets should become stable again according to Theorem 2.3. This however, does not constitute real physical stability as these states cannot be reached from physically stable states since they need to pass through the physically unstable ones. Furthermore these sudden stable regions appear to be caused by the shape bifurcations discussed earlier; when a droplet needs for form additional "pendants" for a short time, the surface area needs to be increased without gaining much gravitational potential energy. This would constitute a form of meta-stability near such bifurcations. Yet in reality a droplet would just yield by forming one "pendant" which would

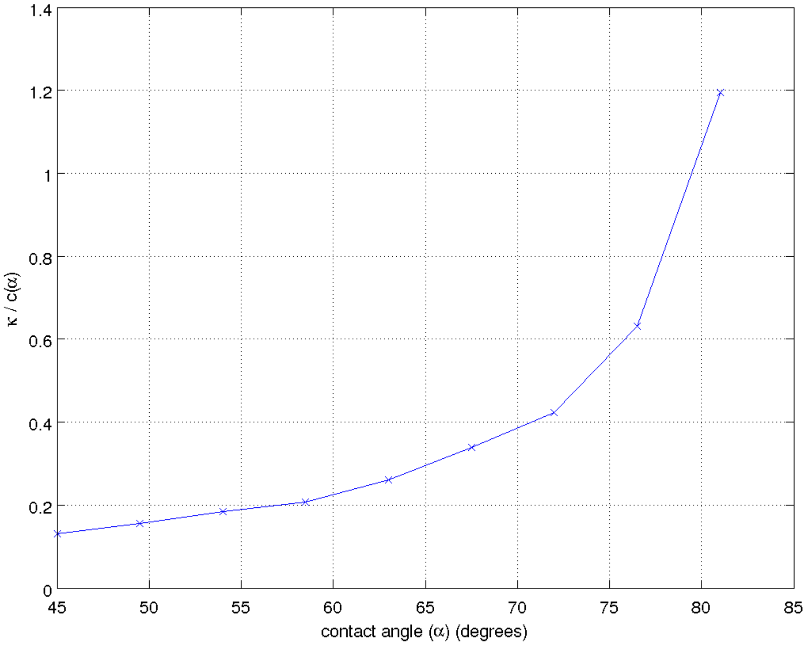


FIGURE 2.7: Plot of $\frac{\kappa(\alpha)}{c(\alpha)}$ in Eq. (2.3.27). As of the time of writing, there exists no theory to explain the form of the function.

From this first onset of instability, we can make a prediction about the dependence of contact angle on the yielding radius, this is outlined in Fig. 2.9. We can also see from this data, that as the contact angle increases, the droplets become deeper (relative to their radius) before they fall than their counterparts with low contact angle.

2.4.2 Comparison to Experiments

From Fig. 2.9 we can predict the contact angle, merely by observing the largest droplet radius hanging from the substrate. The maximum, radius observed in the experiments of LAPP is 0.87 capillary lengths, which according to Fig. 2.9 corresponds to a contact angle of approximately 79°.

This is in agreement with direct measurements of the contact angle in LAPP's experiment. One needs to be cautious though. Real world experiments always have a degree of contact angle hysteresis and contact line pinning, whilst such quantities are not included in this static theory as they are variables depending on the dynamics of the droplets.

Of course this is only a verification with *one data point*, and so does not carry much

consequently drip off. Investigation the existence of these metastable regions of many pendants would require the inclusion of the dynamics of a droplet while it moves from one state to another and thus is beyond the scope of this research.

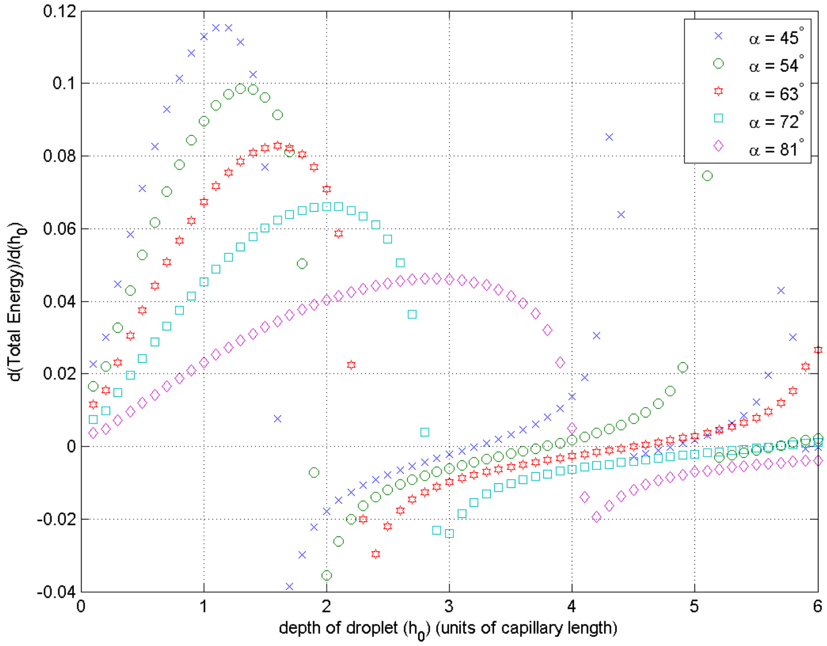


FIGURE 2.8: Plot showing the change in total energy of a droplet given a small positive change in the axial direction. Once this change in total energy becomes negative, the droplet is no longer stable against axial perturbations. The change in energy becomes positive again for very deep droplets, this however is an artifact due to the bifurcations in droplet shape discussed earlier.

weight. Further experiments with different contact angles is left to future research, due to the technical complexity associated with changing the contact angle of the experimental procedures used in tandem to this work.

2.5 Contact Line Forces

Using our data, we may also estimate the force on the contact line with the following lemma:

Lemma 2.4. *Let r be the radius of a droplet on the substrate, given a function $\phi_{r\alpha}$ which minimizes Eq. (2.1.1) according to Eq. (2.1.8), and letting $E(r) = F[\phi_{r\alpha}]$ then the force exerted in "moving" the solution $\phi_{r\alpha}$ to one with a different radius is given by f:*

$$f \simeq \frac{1}{2\pi}\left(\frac{\mathrm{d}E(r)}{\mathrm{d}r}\right)_{\alpha,\, S} \tag{2.5.32}$$

where the derivative is taken while holding the contact angle α and volume S constant.

Proof. Eq. (2.5.32) is nothing more than staying "Virtual work is force times infinites-

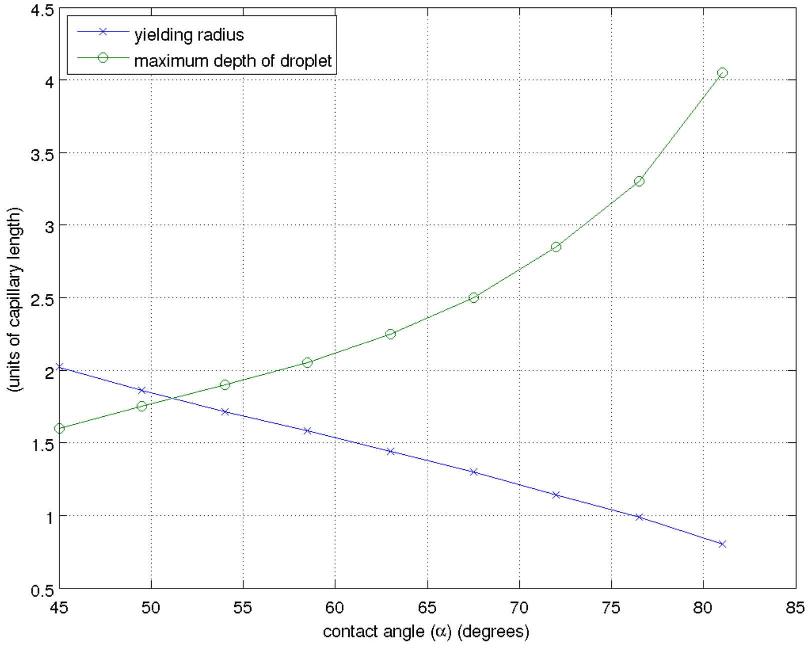


FIGURE 2.9: Plot showing the yielding radius as a function of α (blue line) and the maximum depth of droplets shortly before yielding (green line) for comparison. For larger contact angles, the largest droplets extend further downwards.

imal distance". Thus Eq. (2.5.32) represents the change in the energy given a change brought about by moving the contact line for a small amount which is exactly the definition of a force. □

Lemma 2.5.32 can be thought of giving a *estimate* for the force per unit length along the contact line at the very least. Applying Lemma 2.5.32 to our data yields the dependence of the contact line forces on the radius of the droplet as detailed in Fig. 2.10(a).

For comparison, two extreme cases for contact angle have been plotted showing that for larger contact angles, the droplets exert a greater force onto the contact line and that this force grows with the radius of the droplet. The first observation is an expected result as contact line pinning decreases with increasing contact angle. The second observation can be understood as larger droplets having greater mass, and so these would "pull" more on the contact line. Whereas the sudden drop in contact line forces occurs at the same radius where droplets start yielding, and so is associated with the instability of the solution. This also indicates the dynamics of droplet yielding; once a droplet becomes unstable, a large amount of the bulk pinches off, alleviating the "pulling" of the droplet bulk on the contact line.

This has also been seen by LAPP in experiments involving glass plates, were the larger droplets where "more spherical" i.e., not subject to as much contact line pinning.

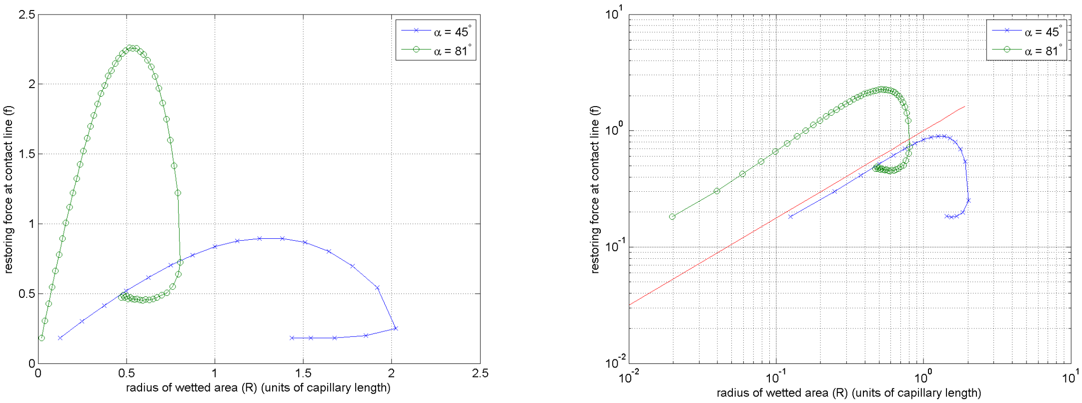
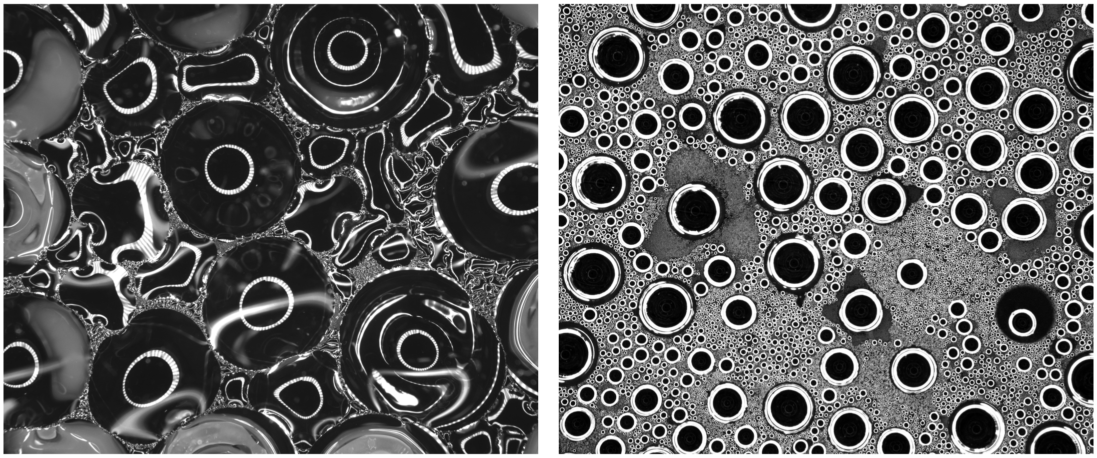


FIGURE 2.10: The force per unit length at the contact line of a droplet with $\alpha = 45°$ (blue line) and with $\alpha = 81°$ (green line). The power law $f(r) = r^{\frac{3}{4}}$ is shown in the red line for comparison. Panel (b) shows the data of (a) on a log-log scale.



(a) (b)

FIGURE 2.11: Pictures from the experiments of LAPP. Fig. 2.11(a) shows the shape of droplets for low contact angles (water on glass). Of particular note is that the large droplets are significantly more round than the small droplets. This is agrees with the maximum observed in Fig. 2.10(a). Fig. 2.11(b) show the shape of droplets on a plastic coating, where the contact angle is significantly higher. We can see that for high contact angles, contact line pinning has little effect.

Fig. 2.11 demonstrates this and that when the contact angle is increased by coating the plate with plastic, then all observed droplets are almost circular.

A final interesting note is the observation that this force is dependent on the radius by a power law of exponent $\frac{3}{4}$ (red line in Fig. 2.10(a)).

2.6 Conclusion

We have extended Wente's model for hanging droplets to account for fixed contact angles. Wente's model has the advantage of being easily applied as solving it merely involves numerically integrating a system of ordinary differential equations.

Our improved model can be applied to axially symmetric droplets hanging off a planar surface with a fixed contact angle less than 90°.

Including contact angle in this model has made it applicable to a wide range of problems. Using this improved model, we were able to determine the dependence of volume on radius for droplets hanging off a planar surface. We were also able to determine the stability threshold of droplets and how this depends on the contact angle. This gives us a means of estimating the contact angle, simply by observing the yielding radius. This was found to be in agreement (at least qualitatively) with experiments. Furthermore, we were able to estimate the contact line forces. This allowed us to explain why an increased contact angle causes droplets to be less sensitive to contact line pinning effects.

Chapter 3

Numerical Implementation

> Hardware: the parts of a
> computer that can be kicked
>
> Jeff Pesis

There exists no characteristic scale for the droplet sizes observed in breath figures, or at the very least, we do not know the scale in advance. This may be illustrated by examining how droplets interact when they produce a breath figure: when the *edges* of 2 or more droplets are in contact, they coalesce. This means that when we wish to determine whether a given droplet is in contact with *any* other droplet in the system, we need to compare it to *every* droplet in the system. Knowing the coordinate of the droplet in question does not help us reduce to number of comparisons that need to be made. This is simply because a very large droplet may be very far away from our droplet in question, yet due to its size, its edge may be very near the edge of the droplet being tested. Hence, storing all droplets merely in an array of N elements would result in the need to perform $N(N-1)$ collision checks for every new droplet added and every coalescence event. Considering that there may be many millions of droplets, this checking may take a long time.

Instead, the more intelligent solution would be to keep a list of "nearby" droplets associated with each droplet. Then our droplet in question only needs to perform collision checking with the nearby droplets without needing to check all droplets.

This does require an intelligent and fast method of updating these lists of nearby droplets.

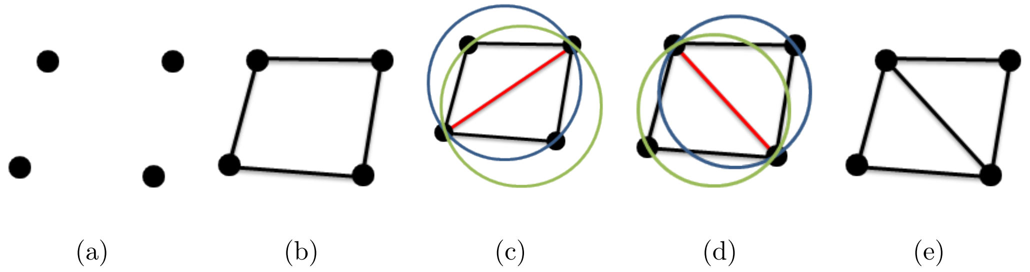


Figure 3.1: The Delaunay property in action: In (a) we have 4 points, for these 4 points, there are 2 possible triangulations. The boundary of our set may be connected trivially as seen in (b). This is the only configuration as links are not allowed to cross. We then have to make a decision about which diagonal to choose. In (c) and (d) we see the two possible choices for the diagonal. For each choice of the diagonal we have 2 triangles which may be circumscribed by a *unique* circle. For the case of (c) at least one node falls *into* into at least one circumcircle. This means that there this troublesome node is closer to the circumcenter. This violates the Delaunay property. However in the case of (d), every circumcircle is empty. Hence the vertexes are either equidistant, or further away, from each circumcentre. Since we are interested in a minimal neighborhood network, we shall choose the configuration (d) as our final choice of the Delaunay network (e). In this way, each circumcircle's center has exactly 3 closest neighbors.

3.1 Delaunay Triangulation

The method of Delaunay triangulations[1] assigns a set of neighbors to each node in a network. This set of neighbors is a minimal set, i.e., any network which assigns each node with fewer neighbors would "miss" some of them.

Furthermore Delaunay triangulations provide themselves as a candidate for this intelligent method for keeping lists of adjacent nodes because it is fast and well studied. As an added bonus, the Voronoi network (which tells us the region closest to each node) is simply the dual lattice of the Delaunay network.

Consequently, in this work we shall then use the Delaunay triangulation to determine.

Appendix A is a detailed background on why triangulations work, instead, here we shall give the reader an intuition into Delaunay triangulations using Fig. 3.1 and the associated caption.

[1] A triangulation of a set of points, is a set of triangles, such that these triangles have the points at their vertexes and no point is left out. In a triangulation the only polygon is a triangle and no edges of the triangles may cross each other.

3.1.1 Computing Considerations

Initially this proved to be anything but a trivial task; most algorithms for finding nearby vertexes in a collection of points are optimized for “sweeping” through all points and constructing the Delaunay and Voronoi network from scratch. We found only little research on locally updating an existing network.

Ultimately a very insightful paper [9] was found detailing a simple algorithm for updating the Delaunay network. This provided us with a strategy for performing our simulations:

1. set up an initial “trivial” grid by hand (connect the few initial points manually using a *A.connectWith(B)* method
2. add a new droplet to a Voronoi cell, connect the new droplet to its neighbors, and locally update the Voronoi and Delaunay networks
3. resolve collisions
4. repeat from step 2, until a criterion for maximum time or droplet size is reached.[2]

It is worthwhile to note that the choice of programming language is very important at this stage. Obviously our algorithm is going to be recursively defined, each object was going to contain a list of arbitrary length denoting its nearest neighbors, and each update may alter an arbitrary number of vertexes in the Delaunay network. These properties make it clear that our algorithm requires:

1. lists of objects with variable length, requiring fast access speeds.
2. a variable amount of temporary memory which cannot be deterministically pre-allocated before it is actually needed (since each update may be followed by an unknown number of nearby updates).

This means that the computer will need to reserve and then free a lot of memory dynamically. Also the language of choice should come with a large collection of methods and objects to efficiently deal with dynamical array lists. As a result the Java programming language was chosen as it has a very efficient and highly tunable garbage collector (to save the physicist from needing to hunt for good memory management algorithms, something which is not really an interesting *physical* problem) and it comes with a large library of generic data types called “Collections” which very efficiently deal with collections of abstract objects which require dynamic memory allocation and require fast access, searching and sorting all the while being large and unwieldy.

[2] or until the computer starts to swap!

This philosophy of “the physics is the interesting part, not the pointers” ultimately resulted in the choice for Java. In particular since for Java 1.2 (and newer versions) the virtual machine performs just in time compilation (the java bytecode is compiled to native code at runtime) which results in program performance which is comparable to that of C++ programs after the Java program has been running for a short time. Furthermore, the JVM does perform some optimization of the code based on information gathered during runtime.

3.1.2 The Mostafavi Algorithm

Mostafavi *et al* [9] proposed an algorithm for locally updating Delaunay triangulations based on “Swap” operations similar to rearrangements in soap foam bubbles. Such algorithms have reported performance between $O\left(\sqrt{n}\right)$ and $O\left(\log n\right)$, for networks containing n nodes, depending on specific optimization techniques for traversing the network [9].

In short, Mostafavi’s algorithm entails:

add Finding a location to add a new node too

walk Performing a “walk” from some previous triangle to the triangle containing the new point P; Fig. 3.2(a)

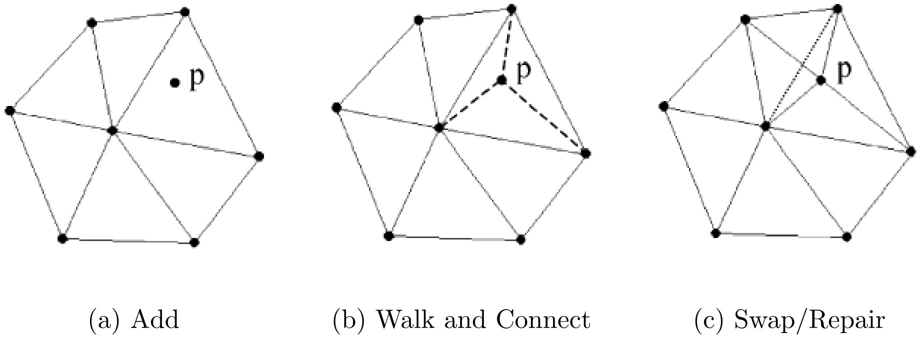
split Splitting that triangle into three new triangles, each having P as a vertex Fig. 3.2(b); and

repair Ensuring that every edge is Delaunay by recursively testing each edge (triangle pair) and switching the diagonal if necessary. Fig. 3.2(c).

Add Algorithm

Fig. 3.2 depicts the general approach to locally adding a node to the Delaunay network. Using a walk algorithm the triangle surrounding the new point is located. Then the point is linked to each vertex within the triangle as shown in Fig. 3.2(b). After the new point P is included in the triangulation, each of the adjacent links is tested if it is Delaunay, and the offensive links are swapped as described in Fig. 3.2(c) and section 3.1.2.

Any updates to the Delaunay network through swapping will require nearby links to be checked also. Thus a recursive definition lends itself as an algorithm that only updates the minimal part of the network. In this case care needs to be taken to avoid stack-overflows.



(a) Add (b) Walk and Connect (c) Swap/Repair

FIGURE 3.2: Sketch of the steps involved in adding a drop to the network. First the triangle surrounding p is identified in Fig. 3.2(a). This can be most readily done using a walk algorithm. Then the point is linked to each vertex within the triangle as shown in Fig. 3.2(b). After the new point P is included in the triangulation, each of the adjacent links is tested if it is Delaunay, and the offensive links are swapped as described in Fig. 3.2(c) and section 3.1.2. Any updates to the Delaunay network will require nearby links to be checked also.

Walk Algorithm

The walk operation uses the determinant $a(A, B, P)$:

$$a(A, B, P) = \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_P & y_P & 1 \end{vmatrix} \tag{3.1.1}$$

If the point $P = (x_p, y_p)$ is to the left of the directed edge AB of some initial triangle, then $a(A, B, P) > 0$. If so, the next clockwise edge from B is used, if not the next clockwise edge from A, and the test is repeated until three successive edges have P on the left, or the right[3]. This is, theoretically, the least efficient part of the algorithm, and more efficient search algorithms may be used for extremely large data sets.

The case $a(A, B, P) = 0$ corresponds to P being on the line AB.

With this algorithm, one may start from an arbitrary triangle and "walk" to a desired point to find the surrounding triangle. This algorithm is the slowest part of performing local updates to a Delaunay triangulation, with typical algorithms performance being reported to be $O(\sqrt{n})$ [9].

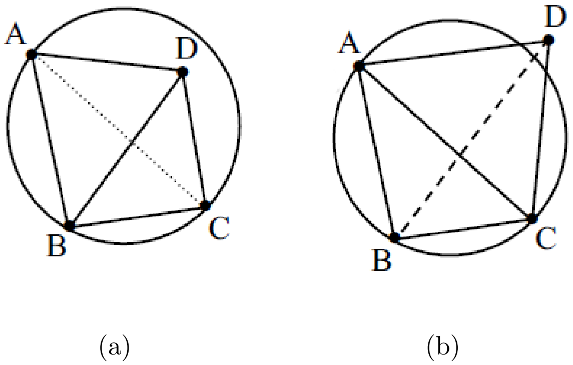


FIGURE 3.3: The swap algorithm explained: In Fig. 3.3(a), the node D is inside the circumcircle for the triangle ABC. Thus the link BD is preferred over the link AC as a triangle ABC would violate the Delaunay property (that all other nodes fall outside the circumcircle of each triangle). In Fig. 3.3(b) the link AC is preferred over the link BD as the point D (which is now further away than in Fig. 3.3(a)) now falls outside the circumcircle for the triangle ABC.

Swap Algorithm

The pivotal idea behind the algorithm is that the Delaunay network is a triangulation. That means every polygon in the network has exactly 3 sides. Furthermore, this network possesses the "Delaunay property", that is every other point in the network lies outside the circumcircle of any given triangle.

When adding new nodes to this network, the new edges might be in violation of the Delaunay property by falling within the circumcircle of a nearby triangle. Consequently, fixing the network proceeds by replacing offensive links by compliant ones in their stead. However, since the Delaunay network is a triangulation, for every link destroyed one must be created, otherwise the number of links is not conserved, creating at least 1 4-sided polygon. We can picture this as turning or swapping the link as shown Fig. 3.3.

In this algorithm, the swap will be the primitive operation. Adding or removing links on their own is syntactically impossible as the data structures were deliberately not given such an operation.[4]

[3] At this point an important correction was incorporated to the algorithm of Mostafavi: the original algorithm suggested only counting left turns, however we have discovered cases where the algorithm would circle around the point P in the right direction.

[4] early implementations of our code did include a `.remove` method, but this caused much more trouble than it was worth.

Delete Algorithm

The deletion algorithm is essentially the inverse to the add algorithm. Thus what needs to be done is to "swap away" all but 3 of the links to a given point, and then remove the last remaining links.

However this is not a trivial task: the nature of the Delaunay object is that every modification to the network results in adjacent links being tested for the Delaunay properties. Hence one needs to first suppress this test for (and only for) the node being deleted (i.e., it is OK that the node being deleted falls within an circumcircle).

Furthermore, one needs to be careful that the swap away operation does not create a new link that crosses an existing link. As this is inevitable when swapping away all of the links for a node, a well defined order for which links get swapped first needs to be established. The most intuitive method is to perform the swap operation which would produce the "least Delaunay" network. Hence the link which would produce the largest circumcircle is swapped into the now offensive position.

We swap the links producing the largest circumcircle first because: when swapping a link into an offensive position, a number of nearby nodes would fall within the circumcircle of the new, now defective, triangle. The network responds in the following way: all nearby links are re-arranged such that the Delaunay property is restored for all points except the one being deleted, essentially, the network "relaxes" into a non-defective state. By swapping the links which produce the largest circumcircles first, this relaxation process has a high probability of swapping many links away from the point being deleted, reducing the number of followup swaps.

We had experienced performance issues with Mostafavi's deletion algorithm as it did not properly test for the newly produced links crossing existing links in cases where deleting a node can potentially produce very long links (as it is the case for removing a very large droplet). The appropriate changes to the algorithm are addressed in the next section.

3.1.3 Modifications to the Mostafavi Algorithm

We modified Mostafavi's algorithm to better deal with (a) adding droplets at random, (b) dealing with periodic boundary conditions and (c) identifying and dealing with droplets that can merge.

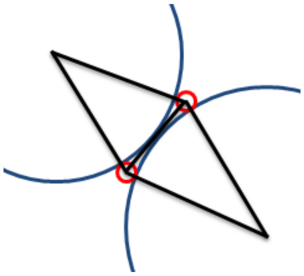


FIGURE 3.4: Case where Delaunay triangulation does not connect adjacent droplets due to a large difference in droplet sizes. This is a *valid* Delaunay network: if the link between the red droplets was swapped, the red droplets would be in the circumcircle of the 2 resulting triangles. This is because the Delaunay network is the minimal adjacency network for centers and not the contact lines of droplets. However, the blue droplets are mutually members of the red droplet's Delaunay networks. We can use this "mutual friends of friends" approach to prevent this fatal overlooking of potential neighbors. This case is unique to systems that have a large range of droplet sizes.

Accounting for Interface-Interface Interactions

In order to digest a subtlety when using Delaunay networks to follow the dynamics of breath figures we consider two very large droplets immediately before they merge. They have two little droplets on either side of the cusp formed between them. Then the Delaunay network might be such that the two little droplet's centers are closer, and ergo there is a link between them. There will not be a link between the two large droplets even though they are in a position to be able to merge, yet the two little droplets are not. Fig. 3.4 is a sketch of this situation. In this case, the Delaunay network exhibits a structure exactly the opposite to what we desire.

It is possible to construct a Delaunay network based on the distance between the droplet's contact lines, however this would be computationally far more intensive than the simple solution we provide here.

In order to cope with this problem, we define "neighbors" and "friends". A neighbor is a droplet whose interface can come in contact with the interface of another droplet without "going through" the interfaces of any others. Hence it is a candidate for merging irrespective of where it is. *This does not have to mean that the droplets are adjacent according to the adjacency matrix of the Delaunay network.* We call droplets friends, if there is a direct link between the droplets in the Delaunay network.

The set of neighbors forms a covering set of all droplets that can merge. Since the set of friends is only a subset of the set of neighbors, we have to keep track of all neighbors rather than going through the set of friends when checking for collisions.

The set of neighbors can be quickly constructed. Since Voronoi cells completely cover the system, the set of friends and friends of friends would form a "ring" of Voronoi cells around the droplet in question which contains all possible candidates which may merge with the droplet in question.

In fact, this procedure can be distilled more so: being a triangulation, 2 friends will also have exactly 2 mutual friends. One of this mutual friends will be the droplet which we want to test for collisions, the other will be the only candidate for collision. Fig. 3.4 provides a sketch of this: the two blue droplets are the mutual friends of the 2 red droplets (who themselves are friends). Hence the set of friends and pairwise mutual friends will be a covering set of all neighbors. Each droplet stores its friends in an ordered `ArrayList<Droplet>`, being ordered by angle to some fixed axis. Hence finding mutual friends requires only accessing the next index and so it is very fast.

Modified Add Algorithm

The integral requirement by our model for the evolution of breath figures is that new droplets are added to the system at uniformly distributed random positions. This way voids are slowly filled up proportional to their size and droplets grow (proportional to their wetted area) in a manner that reflects the phenomenology of nucleation and droplet growth on surfaces.

It would be rather foolish, however, to simply choose a position at random and then walk to this position because the walk algorithm is the slowest step of the Mostafavi Algorithm. Instead a droplet is picked at random weighted by the size of its Voronoi cell as follows:

1. pick a random number r between 0 and A (the system size)
2. pick a droplet, get the size of its Voronoi cell area a.
   (a) if $a > r$, a uniformly distributed random position is picked by picking Cartesian coordinates from the uniform distribution (bounded by the bounding rectangle of the Voronoi cell) and repeating this until it falls within the Voronoi cell.
   (b) if $a < r$, then $r \to r - a$
3. repeat until a droplet is found

This means that ultimately the `.add` operation really means an `.addTo` operation, where a lot of performance is gained because we already know to which node we will be adding the new node after all, the coordinates of the new node are generated such that it falls within the Voronoi cell of the node it is being added to.

A word of warning needs to be said about crossing links: the Voronoi cell might have Delaunay links belong to other Voronoi cells running through it. If the new node falls into an unlucky corner that is being cut off from the center of the Voronoi cell by an external link, then the assumption that it was safe to add the new node to the node which owns the Voronoi cell is invalid! This isn't a problem that Mostafavi had. One might be tempted to now implement the walk algorithm, but that is not necessary! Instead what we can do is "hand off" the add command to the next droplet in the counter-clockwise direction. Eventually the hand offs would reach the perpetrator of the transecting link and we can add the new droplet to this link.

Only when all other means fail, we resort to the walk algorithm, while being slower, is also more robust than more optimized algorithms, furthermore it is independent of boundary conditions.

Periodic Boundary Conditions

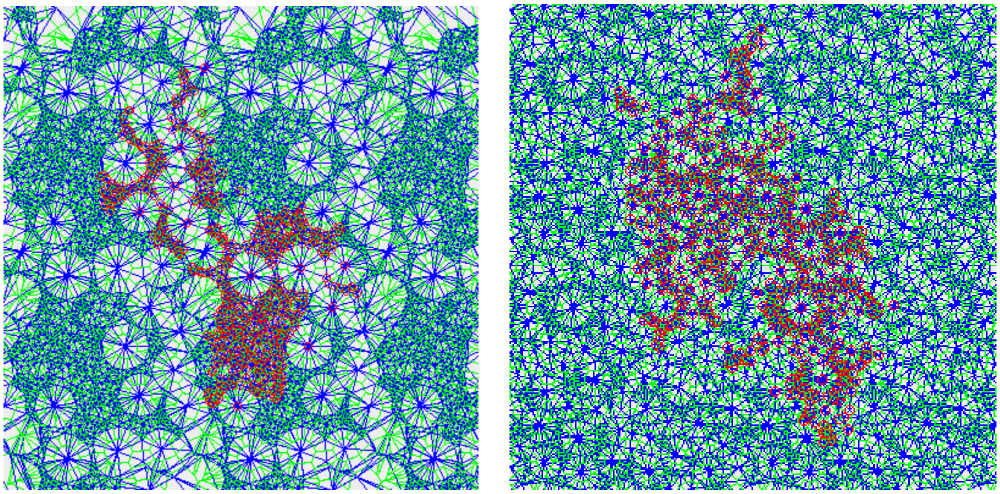
We have employed a simple trick for dealing with periodic boundary conditions: every time a droplet is added to the network, 8 copies are added in the 8 adjacent periodic images. In this sense the periodic images of our system are also real (and have an "edge", however this does not matter as the information from these is not included in the data analysis).

One might ask, *why use such a method? Why not define distances on a torus?* This is a question of pragmatism over idealism. As explained in the section on the implementation of our code, a clear "hierarchy" of information was followed: the Delaunay portion only knows the droplet's positions in real coordinates, then a class for periodic boundary conditions was extended from this. In other words, the individual nodes of our network should not "know" that they are actually living on a torus. This allows the code on Delaunay networks to be applied to many different problems not just those involving problems derived from periodic boundary conditions.

Care needs to be taken when implementing the periodic "copy" approach, as the way in which the add algorithm works can produce severely elongated circumference and aspect ratio, as shown in Fig. 3.5.

This would cause the program to eventually crash as 8 periodic copies on a square grid are not enough to surround the central period if the system becomes so contorted that its length is comparable to 2 times the length of an original period.

As a consequence, bias is incorporated in to the add algorithm; such that when an added droplet falls outside a bounding box of twice the system size, we try to move the droplet to the corresponding place on the other side of the system. This does not always work because of errand links being in the way and so some problematic droplets



(a)

(b)

Figure 3.5: Voronoi (green) and Delaunay (blue) diagrams for breath figures. The periodic boundary conditions have been displayed explicitly. Red circles denote the droplets belonging to the list of droplets in the "central" period. While certainly being very pretty, the unusual pattern of the central period ultimately becomes pathological since the dynamics of our simulations might cause their width to reach the other side of one of the adjacent units.

remain outside the bounding box.[5] As we can see from Fig. 3.6 such a bias is sufficient to produce periodic boundary conditions without contorted systems.

One particularly beautiful feature of periodic boundary conditions is that the sum of the areas of all the Voronoi cells needs to equal the system size. A special cautious mode has been implemented which regularly goes through all droplets and measures the sum of the Voronoi cell areas. The program would terminate if this did not equal the predefined system size. Also whilst saving, the total size of the Voronoi cell areas is measured (as the program would have to read the entire memory anyway), and an error would be printed in the output if the data ought not to be trusted.

Modified Delete Algorithm

As described in the previous section, the delete algorithm had to be modified to account for the new link crossing other already existing links. This is only an issue when the Voronoi cell of the node being deleted is very large and has many neighbors. In this configuration a link originating from a neighbor's-neighbor's-...-neighbor could almost

[5] As a passing note: it is not always possible to "move" a droplet from one of the periodic copies because: 1) the periodic copies are "watered down" objects referencing only the most relevant information from the parent, and 2) droplets are stored in unsorted data structures, one might not be able to find the droplet's virtual copies in memory with reasonable computational effort.

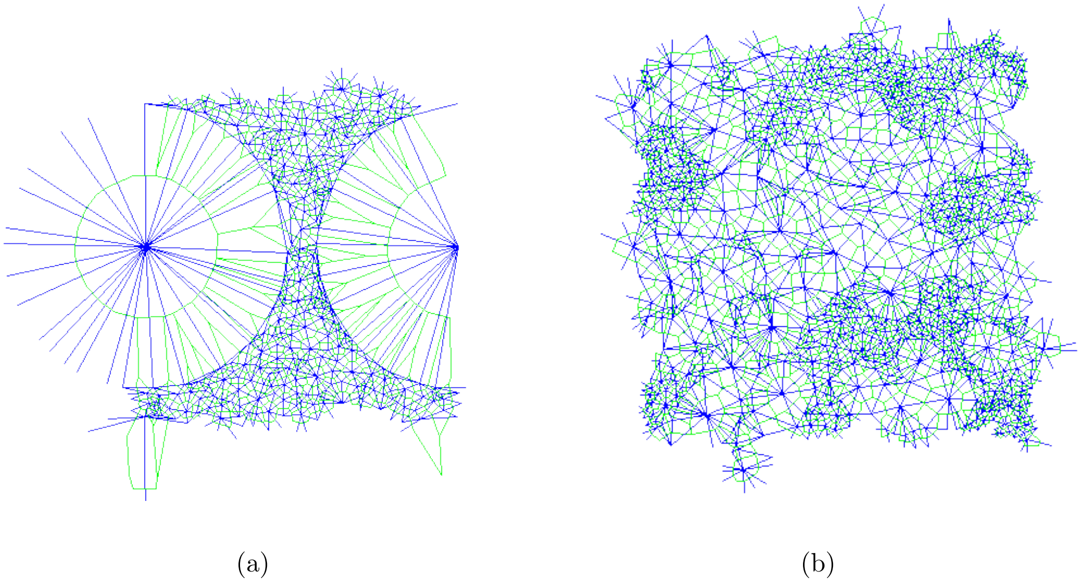


FIGURE 3.6: Voronoi (green) and Delaunay (blue) diagrams for the fundamental domain of breath figures in a system with periodic boundary conditions.Fig. 3.6(a) shows the result for a system containing a massive droplet. Fig. 3.6(b) shows a steady state of a system with gravity

cross half of the freed region. And so the algorithm does unfortunately have to check all internal links of the region being freed for possible intersections. Mostafavi's algorithm only checked *direct* neighbors for intersecting links and therefore produced nasty errors whenever very old droplets were removed.

The algorithm goes as follows:

1. Set the `private boolean is_alive` flag to `false`. This will prevent the node from being considered in Delaunay testing.[6]

2. Go through all adjacent nodes and determine the radius of the potential circumcircle. Then swap the link producing the largest circumcircle. If this link crosses any other links, the swap is undone and the next largest circumcircle producing link is swapped.

3. Repeat step (2) until only 3 links remain. Then remove these 3 links directly by erasing the node.

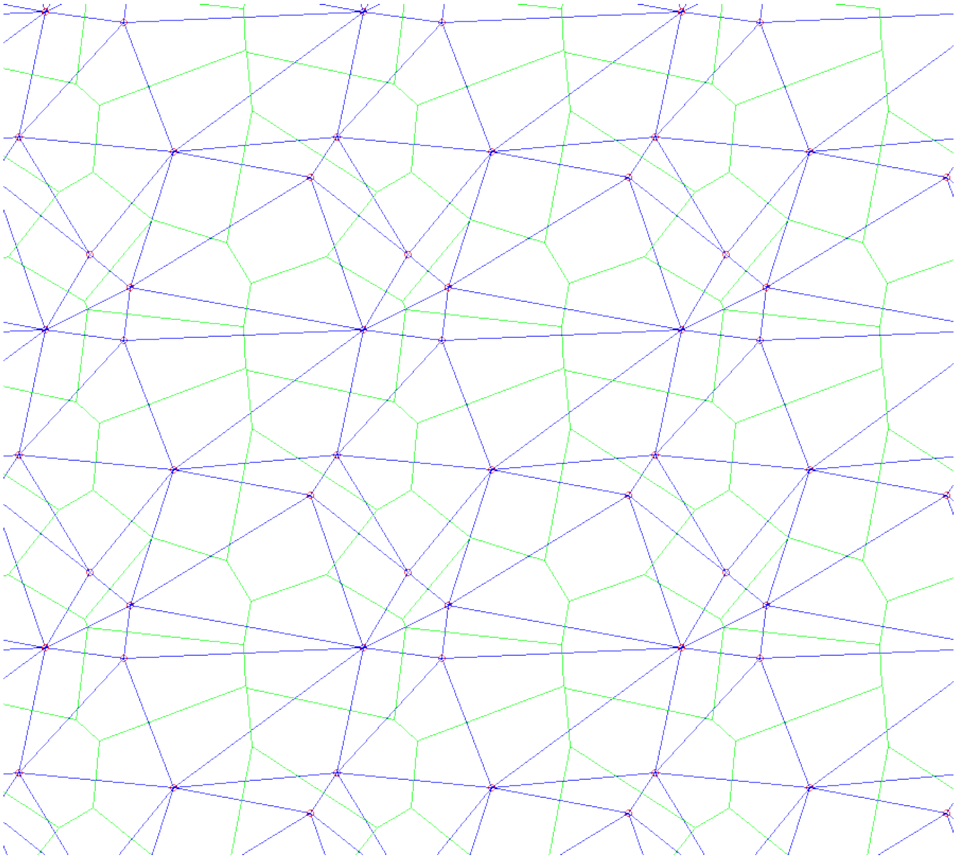


FIGURE 3.7: The Delaunay and Voronoi diagrams for a set of points (red circles) for a fairly evenly distributed set of points in periodic boundary conditions. The Delaunay Triangulation is the blue network (notice that this network is indeed a triangulation). The Voronoi diagram is the green network with vertexes at the centers of the Delaunay triangulation's circumcircles.

A Common Misconception

The most common misconception is that the Voronoi-cell lines have to actually intersect with the Delaunay grid. While the Voronoi network is the dual network to the Delaunay network, and thus does not have to bisect every edge of the Delaunay network, these are very frequently depicted as bisecting each other. One may conjecture that these pictures are frequently used due to their aesthetic value. Fig. 3.7 is a good example of a "real" system where either network rarely bisect. The implementation of this algorithm therefore was a good example how literature can shape ones subconscious assumptions and only an examination of the basic geometric properties (*without* pictures) made this grave mistake apparent whilst frustratedly attempting to convince the computer that certain lines *should* intersect.

[6] The object oriented nature of our code means that every modification automatically results in Delaunay testing.

3.2 Triangulation-Based Algorithm for Simulating Breath Figures

The philosophy that the triangulation ought to be separate from the code which deals with periodic boundary conditions — and that this should in turn be separate from code which deals with droplets — is very appealing and thus dictated the shape of our program.

All code relevant to triangulations was implemented in the class[7]

```
public class NetNode<extends NodeContainer>
```

With this philosophy at heart, all code relevant to updating and maintaining the triangulation is in the `public class NetNode<extends NodeContainer>`. We then extend this class with

```
public class PeriodicNetwork<extends NodeContainer> extends NetNode
```

to deal with periodic boundary conditions. Leaving the triangulation code unharmed and applicable to many other uses if need be. The periodic network is then extended upon by

```
public class Plate<extends NodeContainer> extends PeriodicNetwork
```

which deals with the fact that we are simulating droplets which interact upon interface contact and are distorted by gravity.

We then use the Delaunay network for storing the droplets in memory. There exists no grand register of all droplets, instead they are stored in the heap and contain pointers to their neighbors given by the Delaunay network.

As part of an investigation into which droplets fall off first, each droplet is assigned a "date of birth" upon creation. When 2 or more droplets merge, the resulting droplet's date of birth will be that of the oldest droplet, this is almost invariably the larger droplet. The age of droplets may then simply be computed by taking the system's age and subtracting the date of birth. If a droplet falls off the substrate due to excessive mass, its remnant is given a new date of birth corresponding to the current system age.

3.2.1 Implementation

The implementation of the algorithm described in the previous section, along with the modifications we have had to make, was given the name "Doogle" (read: Droplet-

[7] <> denote Generic Data types. That's right! Everything that extends the container object `public class NodeContainer` can be put onto each node. This is very useful as it allows us to include additional properties in the droplets without needing to modify existing code.

Google) in honor of the fictional character Father Dougal McGuire from the television comedy “Father Ted”. The final implementation relies on a very large amount of (unexpected) special cases and complex logic, this made the first versions intractable to debugging and so, our program is deserving of a name reflecting the extremely dim-witted but well intentioned Dougal McGuire.

Difficulties

As stated in the previous section, the slowest step of Mostafavi’s Algorithm is the walk procedure for finding the correct triangle which is to be updated. As we already know the Voronoi cell which to add the new network vertex too, it seemed logical to make the modifications described in section 3.1.3.

However, for many technical reasons, this kind of algorithm requires many special cases to work. The most common may be described as the `public boolean is_complete` property of every network node. If `is_complete` is `true`, then the node has sufficiently many nearest friends surrounding it for there to exist a finite[8] Voronoi cell. This would then allow for many simplifying assumptions. In the case of `is_complete` being `false` a node might not have a surrounding triangle, consequently a different set of rules needs to exist for such droplets.

Furthermore, since the Delaunay network is also the structure of our memory, over-connectivity in the regions not used to obtain physical data is better than under-connectivity. As a consequence, algorithms which distinguish between edge nodes and interior nodes had to be developed (and their bugs “ironed out”). An important lesson was that it is OK for Delaunay links to cross at the edge to prevent isolated “peninsulas” from forming (these would slow down the code). Of course identifying if a node is at the edge of the system is non-trivial!

In the end the following approach was used: attempt to use our optimized code and if this fails (because of a special case), the less optimized but more robust code is employed.

Computational Considerations

In order to simulate the growth of breath figures, the numerical implementation discussed in section 3 was executed on a AMD64 architecture with 30 GB of RAM. Depending on the system size, and whether yielding was allowed to occur, the programs ran between 2 days and 2 weeks. Systems with yielding could theoretically be allowed to ex-

[8] *How can there be an infinite voronoi cell*? Recall that periodic boundary conditions which force all Voronoi cells to be finite are defined in an extension class: the nodes of the Delaunay network live in Euclidean space! Hence the edges of the periodic images actually have fewer neighbors, hence their Voronoi cell can become infinite.

ecute indefinitely, but were ultimately terminated once limited disk space prevented the saving of new data. This was done manually and not subject to any criterion. Whereas, systems without yielding where configured to purposefully halt once the largest droplet's radius exceeded 10% of the system's size.

In all simulations, dimensionless units where used with a minimum droplet size of $s_{min} = 1$. All length scales are expressed in term of the smallest droplet size s_{min}.

In order to speed up the numerics, the volume of droplets did not include the coefficient $c(\alpha)$ (defined in Eq. (2.3.27)), i.e., we chose α such that $c(\alpha) = 1$. Solving Eq. (2.3.30) yields: $c\,(59.7°) \simeq 1$. And so all simulations may be thought of as directly corresponding to breath figures where the liquid has a contact angle of 59.7° on the substrate.

The time scale τ was chosen in terms of droplets added[9] per unit area. For a system with period l, the total area is l^2, and thus our time scale will be $\tau = \frac{N}{l^2}$, where N is the number of droplets in total that have been added to our system. This time scale is very intuitive because when τ is near 1, then every position in the system should have had about one droplet placed on it.

While not being an original problem as such, the case of sessile spherical-cap droplets was still considered, principally to serve two goals: a) serve as a verification of the code's correctness by comparing these results to those of Family and Meakin and b) gain further insights into the "original" theory of breath figures as our code was significantly more advanced than that of Family and Meakin allowing for more information to be gathered of the system.

3.3 Conclusion

Here we have developed an efficient algorithm for following the evolution and growth of breath figures. This code allows us to track merging and yielding of droplets on a planar surface.

In order to determine the nearest neighbors, we have implemented Delaunay triangulations. Whilst typical applications of Delaunay triangulations in physics are based on efficient algorithms which take sets of points and calculate the triangulation networks from scratch, our application called for a network which can be efficiently updated. To this end, we modified the algorithm of Mostafavi [9]. This new algorithm has the advantage that only very few collision tests need to be made whenever a droplet is added to the system, or whenever droplets coalesce. More transitional algorithms would require

[9] As we will discuss later, the number of droplets added to the system is equivalent to a time scale (taking the metaphor of rain further: if the rate at which rain droplets fall to the ground is constant, then one may estimate the time simply by counting the number of rain droplet impacts)

$N(N-1)$ collision tests for systems with N droplets. In order to deal with this diverging number of tests, the algorithms of Family and Meakin and Briscoe and Galvin would rescale droplet sizes, achieving a large range of droplet sizes without requiring many "new" droplets to be added to the system. Since we faithfully follow the physical growth mechanism by adding only droplets of the same minimal size s_{min}, our code is an improvement over that of Family and Meakin in that it more closely reflects reality.

The programming paradigm that the different attributes of the model system ought to be separated in independent classes allowed for easily extensible code which is not only applicable to just one particular situation. That is, the interaction model, boundary conditions and triangulation processes are all separate features and thus ought to be able to exist separate from each other. As a result, the means of calculating the Delaunay triangulations and the boundary conditions may be freely changed without requiring a review of the whole code. Similarly, further physical processes can extend the model system easily. For instance, the droplets may be given additional features, and even physical processes following different time scales may be introduced (such as Oswald Ripening) without needing to modify existing code. This was particularly useful, since the investigation in section 5.1.4 on the origin of the bump[10], in the size distribution of sessile droplets, was conceived *after* the first simulation results had been analyzed. If more traditional coding practices had been used, the fundamental properties of the data structures might have needed augmentation to allow for the droplet age. This would have resulted in a lot of tedious work. However the usage of generics allows the droplet object to carry any kind of data, ergo our modifications were only necessary on the highest level.

[10] the bump at the position $S(\tau)$ in Fig. 1.4(b).

Chapter 4

Sessile Droplets

> Science is what we understand well enough to explain to a computer. Art is everything else we do.
>
> Donald Knuth, *The Art of Computer Programming*, 1969

In the previous chapter, the problem of the shape of a *single* hanging droplet has been examined. Here we shall now examine systems of *many* droplets residing on a two-dimensional surface.

4.1 Model

Family and Meakin introduce a simple model in order to simulate the growth and coalescence of breath figures:

1. Pick a coordinate in the system at random (in the case of this work this will always be a uniform probability density function, representing homogeneous nucleation).
2. Place at this coordinate a droplet, henceforth denoted: n, of fixed initial size s_{min} located at the position $\vec{q}_n$.
3. Determine whether this new droplet overlaps with its neighbors and merge droplets if appropriate:
   (a) Find the closest neighbor, henceforth denoted: m, with size s_m located at position $\vec{q}_m$.

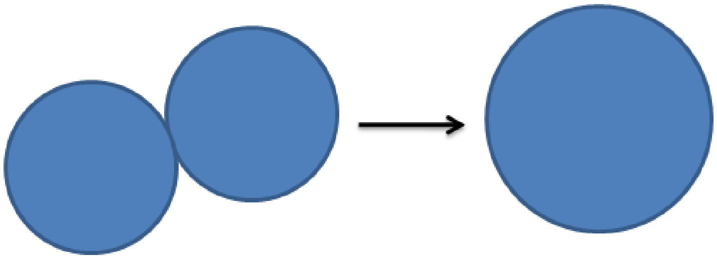


FIGURE 4.1: When two droplets overlap or touch each other, then they merge into a new droplet whose size is equal to the sum of the original droplet's sizes and whose position is the center of mass of the original droplets.

(b) If the droplet n at position $\vec{q}_n$, overlaps with droplet m: merge them as depicted in Fig. 4.1:

(c) Remove the droplets n and m and place in their stead a new droplet, denoted: r, with mass $s_r = s_n + s_m$ at position $\vec{q}_r = \vec{q}_n \frac{s_n}{s_r} + \vec{q}_m \frac{s_m}{s_r}$

(d) Repeat these steps for droplet r until no further collisions occur

4. Repeat all steps until the program is terminated.

Initially one might come to the mistaken conclusion that this model ignores one obvious physical process found in nature: that droplets grow not only through coalescence but also by absorbing material from the surrounding over saturated vapor. This is an erroneous conclusion to come to because when new droplets are added in step 2, this position might already be occupied by a droplet. The result would be that a merger between the existing and the new droplet immediately follows. As this probability of an existing droplet being "hit" by a new droplet is proportional to the droplet's wetted area, one may consider this model as having a droplet growth rate proportional to the droplet area.

Thus, an appropriate metaphor for this model is that of rain: new droplets of a fixed size s_{min} "rain" onto a substrate already occupied by a collection of droplets. This shall be the minimal model used throughout this work. The addition of further physical phenomena such as Oswald ripening, inhomogeneous nucleation and different droplet growth rates are beyond the scope of this work, yet will frame the focus of future research.

4.2 Numerical Simulations

The system studied has periodic boundary conditions, with a period of 1000. During the simulations, the collection of droplets was stored after every 10000 droplets added.

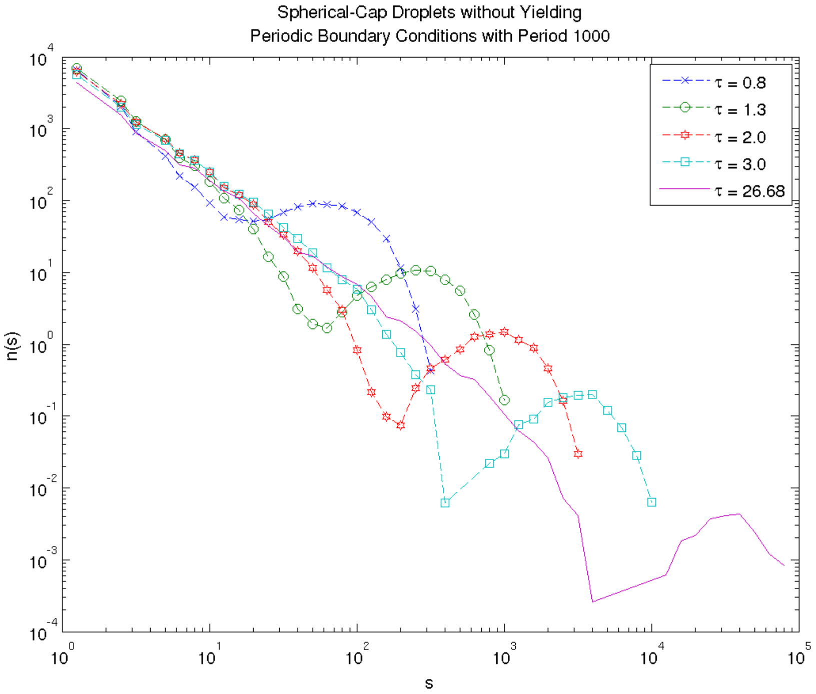


FIGURE 4.2: The size distribution of sessile droplets in a system without yielding nor shape distortion due to gravity. $n(s)$ is the number of droplets with a size s

4.2.1 Results

Fig. 4.2 shows the number of droplets $n(s)$ of size s observed in our system at time τ when no yielding, nor shape distortion due to gravity, is present. The time evolution is also presented.

We can clearly see several important features:

1. The number of droplets of a small size follows a power law distribution, $n(s,\tau) \simeq s^{-\theta_n}$ for some positive scaling exponent θ_n.
2. There appears to be some non-power law behavior for larger droplets, where there appears to be an abnormally large quantity of large droplets. We shall call this the "bump".
3. The power law region of the distribution is independent of time, whereas the bump moves to larger sizes as time increases.

These qualitative observations agree with those of FAMILY AND MEAKIN [8] observation 2 is highly indicative that there exists a characteristic size in our system, and that this characteristic size is time dependent. As the bump is centered around the largest droplets in the system, the characteristic size, $S(\tau)$, clearly is highly dependent on the

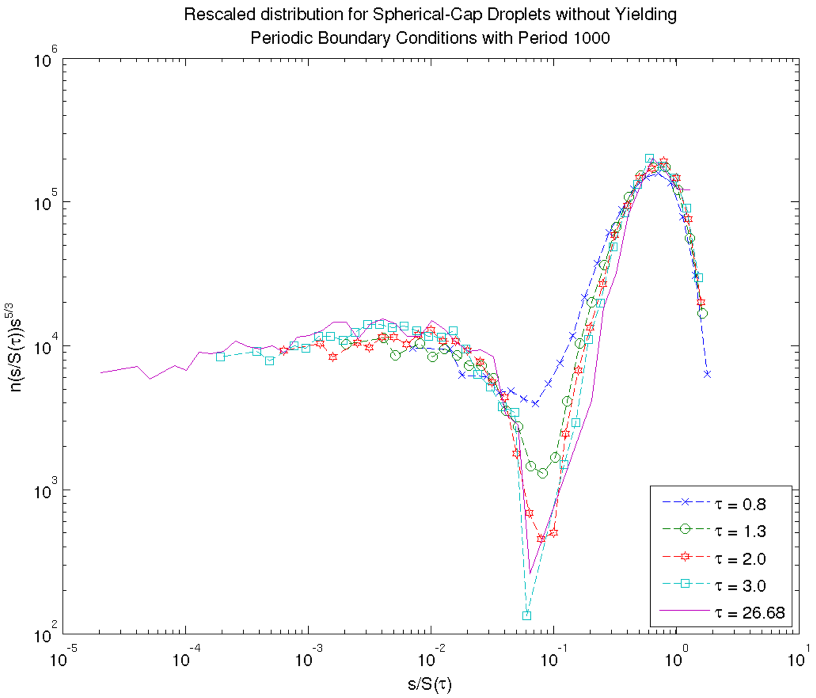


FIGURE 4.3: Rescaling the droplet size distribution Fig. 4.2 according to Eq. (1.2.1a) will yield the scale-invariant function $f(x)$.

largest droplets, where the smaller droplets provide a "background". Furthermore observation 1 indicates that, provided we remain within the domain of relatively small droplets, the observed size distribution is invariant to rescaling the size scale of our system. This scale invariance is not effected by time. However, as time progresses, the region of size scale invariance increases depending on the size of $S(\tau)$.

Based on these observations FAMILY AND MEAKIN first postulated that there must exist some data-collapse which allows for the results to be rescaled into some scale-and-time invariant curve $f(x)$ i.e., they postulated the following self similarity law:

$$n(s,\tau) = s^{-\theta} f\left(\frac{s}{S(\tau)}\right) \tag{4.2.1}$$

Where $f(x)$ is a function that is "fairly flat" for small x, and then exhibits some sort of cut-off, followed by a peak reflecting the presence of the bump. FAMILY AND MEAKIN's argument is detailed in section 1.2 and suggests that the exponent θ should be $\frac{5}{3}$.

We may graphically measure the scale-invariant function $f(x)$ by rescaling our data according to Eq. (1.2.1a). The results of this are displayed in Fig. 4.3 .

As all of the distributions displayed in Fig. 4.3 appear to fall onto one single curve, we can say that the simulated system does exhibit scale invariance. Furthermore, we now have a graphical representation of the universal function $f(x)$. One may phenomenologically

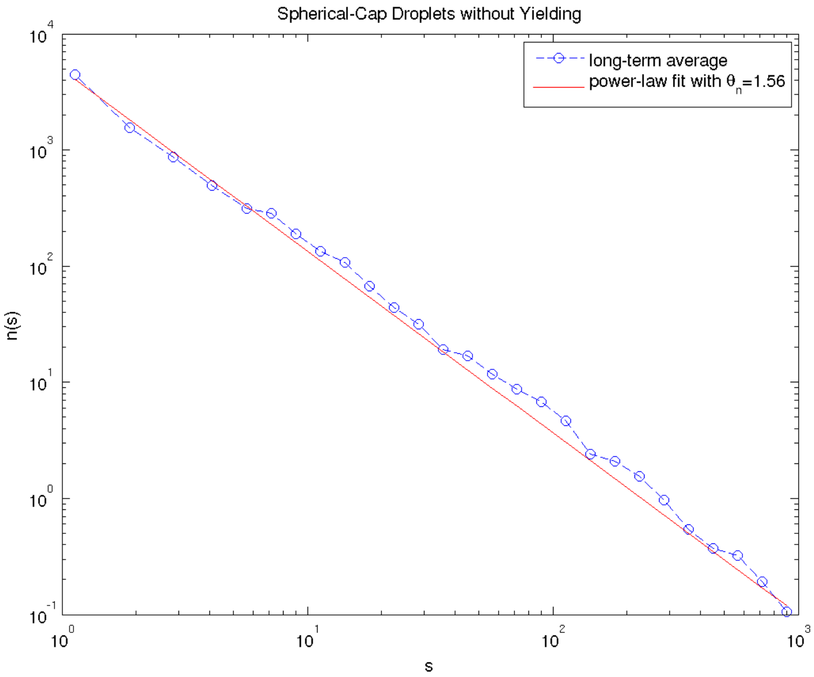


FIGURE 4.4: Examining the power-law distributed portion of the oldest size distribution displayed in Fig. 4.2. Applying a Levenberg-Marquardt non-linear fitting algorithm to the data yield an estimate for the scaling exponent being -1.56.

express $f(x)$ as follows:

$$f(x) = x^{\delta} g(x) + h(x) \tag{4.2.2}$$

where $g(x)$ is a constant for $x \ll 1$ and exhibits a cutoff close to $x = 1$, and $h(x)$ is a function which is zero everywhere except for the bump centered about $x = 1$.

What is also of interest now is the scaling exponent θ_n, where θ_n is the exponent of the size distribution $n(s)$ for the power-law region. From Eq. (1.2.1a) and from Eq. (4.2.2) we know that $\theta_n = \theta - \delta$. In other words, δ provides a "scaling correction".

Fig. 4.4 shows the power-law region for the oldest size distribution displayed in Fig. 4.2 with a power-law fitted to the data. Using the fitting method described in section B (this being used in all other parts of this work henceforth), we find that the power-law exponent θ_n is 1.56 which shows that the predicted exponent needs to be corrected by $\delta = 0.11$.

As we will discuss in section 4.3, there are further quantities of interest in this system. Two such quantities are the number of droplets $N(\tau)$ in the system and a characteristic cluster size $S(\tau)$ which was introduced earlier in Eq. (1.2.1a). The characteristic cluster

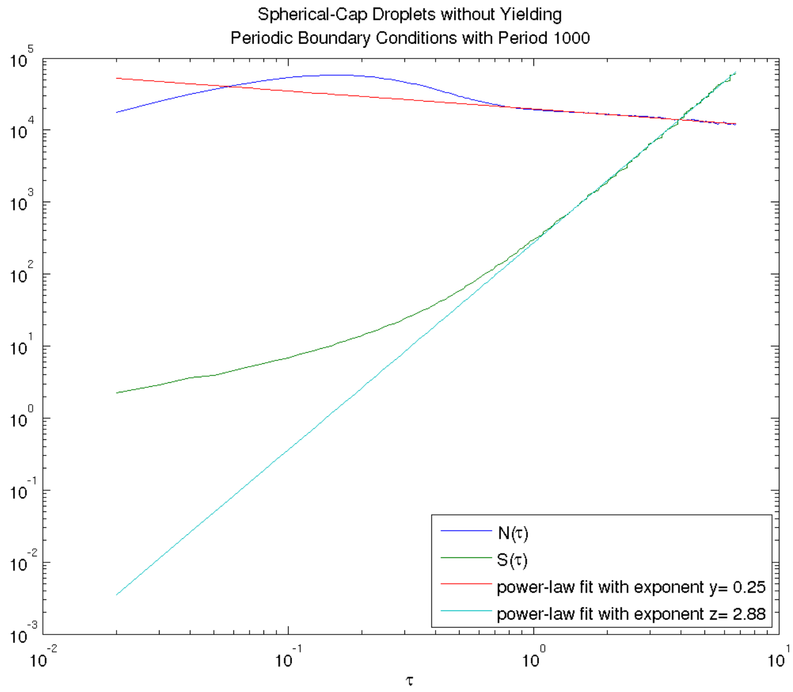


FIGURE 4.5: The number of droplets $N(\tau)$ (blue line) and the characteristic size $S(\tau)$ (green line) plotted over time. Power laws were fitted to the long term behavior of these quantities showing good asymptotic agreement with power law behavior

size was defined by FAMILY AND MEAKIN to be:

$$S(\tau) \equiv \frac{\sum s^2 n(s,\tau)}{\sum sn(s,\tau)} \tag{4.2.3}$$

which is a length scale that strongly favors the largest droplets in the system. The quantity $S_q(\tau)$

$$S_q(\tau) \equiv \frac{\sum s^q n(s,\tau)}{\sum s^{q-1} n(s,\tau)} \tag{4.2.4}$$

only stabilizes for $q > 3$ indicating that if we chose $q = 2$, as was the case with FAMILY AND MEAKIN, then $S(\tau)$ would represent a size slightly smaller than the largest droplets. Furthermore the data-collapse shown in Fig. 4.3 is best if $q > 3$. Hence we shall choose $q = 5$ for all data analysis henceforth.

Fig. 4.5 shows the time evolution of the number of droplets contained in the system, $N(\tau)$ and the characteristic size $S(\tau)$, both showing good agreement with an asymptotic power-law behavior. Especially the characteristic size $S(\tau)$ is required to follow a power law for the scaling theory described in section 1.2 to hold. The number of droplets is expected to follow a power-law behavior in the long term [8].

An important consistency check is given by the total mass (i.e., total volume) contained in the system. As we do not allow droplets to leave the substrate, and since new droplets

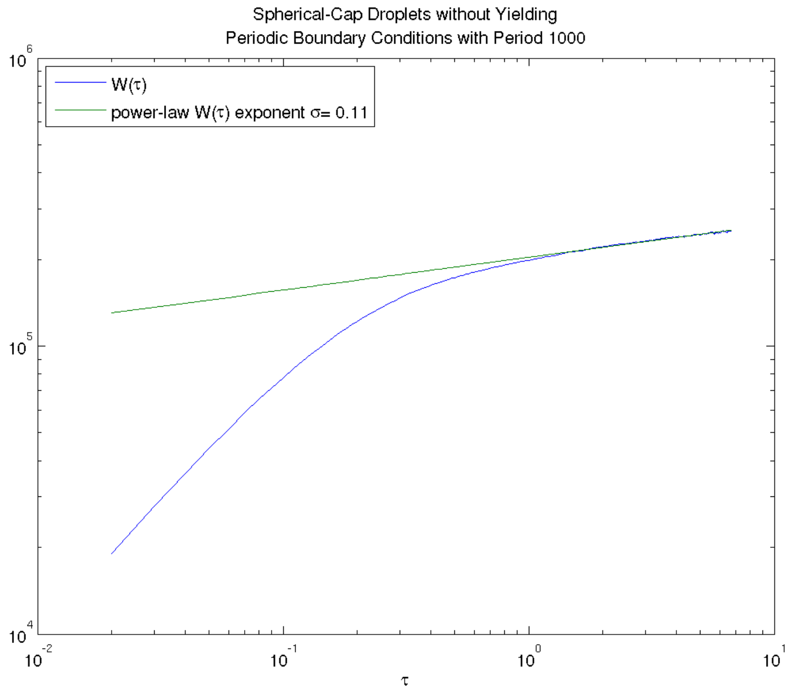


FIGURE 4.6: The total mass in the system $M(\tau)$ (blue line) and the wetted area $W(\tau)$. Fitted power laws show that asymptotically $M(\tau) \sim \tau$ and $W(\tau) \sim \tau^{0.11}$, however, there might be a plateau for extremely long time scales. The simulations were terminated too soon to conclusively determine the existence the this plateau (marked in purple)

have a fixed volume, the total mass, $M(\tau)$ will be proportional to the time scale τ. Hence we would expect that $M(\tau)$ is a power law with exponent 1. This is indeed the case.

Furthermore, the scaling theory in section 1.2 requires that the wetted area remains constant over time for the long term limit. This does not appear to be entirely the case, as a very small time dependence is apparent from the data. However it is also not clear whether this time dependence will persist, or whether it will reach a plateau in the extremely long time limit. Unfortunately the simulations halted too early to be able to determine whether a plateau of constant wetted area will ever be reached. As a constant wetted area is a natural consequence of the self-similarity ansatz Eq. (1.2.1a), if there is no final plateau for the limit $\tau \gg 1$ then the system would not be self-similar. Furthermore, one would eventually expect that the wetted area reaches a constant value because the system has only a finite size and eventually it would be full. Of course, then the sensible question is *does the wetted area reach a constant before finite size effects play a role?* Or does it follow a power law for some time until the system is filled up. Fig. 3.6(a) shows a simple example for why the wetted area should not reach the system area before finite size effects dominate, since in Fig. 3.6(a) finite size effects clearly dominate, yet it is not full either.

4.2.2 Consistency with Family and Meakin

The model system consisting of non-yielding sessile droplets of spherical-cap shape has already been studied by Family and Meakin and so provides us with an excellent method to check the consistency of our simulations against theirs. Family and Meakin did not implement the model in section 4.1 in the same manner as we did. In the simulations of Family and Meakin, simulations were conducted in such a manner that all existing droplets were rescaled after each new droplet had been added. This has the advantage that one may cover a larger range of sizes in comparatively little simulation time. However, this method assumes that the scaling ansatz in Eq. (1.2.1a) is correct in the first place. We on the other hand have not performed rescaling during our simulations. A droplet of size Ns_{min} would have at one time consisted of N individual droplets (not necessarily at the same time). While this requires a significantly longer computational time to reach the same sizes the simulations employing rescaling, our results can be used to confirm that the scaling ansatz (1.2.1a) is in fact valid. Furthermore, we can test to what extent deviations from the scaling exponents are due to physical processes and not artifacts generated by computational tricks such as rescaling.

Family and Meakin report a scaling exponent of $\theta_n = 1.54$, this is within the 1–σ confidence interval (refer to Appendix B) of our scaling exponent $\theta_n = 1.56$. Furthermore, the scaling exponent for the time dependence of the characteristic size is reported to be $z = 2.92$. This lies outside the confidence interval of our result of $z = 2.87$. This discrepancy can be explained, by the fact that we utilized a somewhat more convenient characteristic size function (giving better a better data-collapse) and that our method of simulating did not employ rescaling. If the size distribution was to be rescaled during the simulation, then it stands to reason that the time dependence of the characteristic size ought to be closer to the theoretical value predicted by the scaling theory (in the introduction). Our scaling exponent for the number of droplets[1] is $z' = 0.23$, and so Family and Meakin's result of $z' = 0.26$ lies beyond the confidence interval. As with the case for z, the reason is most likely also linked to our method of simulating. Nevertheless, while not in good agreement, our results for z and z' do not contradict the previous works. Consequently, our implementation of the droplet coalescence model has been successfully validated.

The scaling exponents for the total mass $M(\tau) \sim \tau^{1.00}$ and for the total wetted area $W\tau \sim \tau^{0.11}$ have not been reported in literature. That M is (effectively) linear with time is an important verification that the program is correctly implemented, otherwise does not have much use. Whereas the W not being entirely constant with time is an important realization as it indicates a (minor) violation of the self-similarity ansatz Eq. (1.2.1a) suggested by Family and Meakin.

[1] Family and Meakin observe that the total number of droplets appears to follow $N(\tau) \sim \tau^{-z'}$ [8].

Only simulations which extend the time scale will be able to shine light on this question. Since these are still being conducted, a definitive answer cannot be given at this time.

4.3 Scaling Arguments

4.3.1 Scaling Theory with Finite Quantities

Finite size effects due to a finite system size were ruled out simply by conducting simulations for different system sizes and determining that the scaling exponent does not change significantly. Since power-law distributed data is very sensitive to its lower-cutoff, we notice that the proof for Theorem 1.1 does not explicitly include the size of the smallest droplets.

Thus we arrive at two principal causes for the deviation from scaling theory:

1. As the size distribution is largely a power-law distribution, its normalization is strongly dependent on the smallest droplet size. In essence, Family and Meakin's theory may be improved by accounting for the boundaries of their integrals in the proof to Theorem 1.1.

2. It seems clear from simulations is that the wetted area does *not* asymptotically go to a constant value. Instead it rises according to the power-law: $W\tau \sim \tau^{0.11}$ according to Fig. 4.6.

Theorem 4.1. *Given a system with d-dimensional substrate and D-dimensional droplets, without droplets yielding, and subject to mass conservation: Eq. (1.2.2a), and where the wetted area, Eq. (1.2.2b), follows $W(\tau) \sim \tau^{\sigma}$. Assuming that the scaling ansatz: Eq. (1.2.1a) and Eq. (1.2.1b) still holds with with number of droplets present in the system being subject to: $N(\tau) \sim \tau^{-y}$ then:*

$$\begin{aligned} z\delta &= y && (4.3.5)\\ z(2-\theta) &= 1 && (4.3.6)\\ z(1+d/D-\theta) &= \sigma && (4.3.7)\end{aligned}$$

Proof. We start from the normalization of our size distribution, and apply our scaling ansatz, however this time we pay proper attention to the limit of our integrals. Let ϵ be the size of the smallest droplets, then:

$$N(\tau) = \int_{\epsilon}^{\infty} n(s,\tau)\,\mathrm{d}s = \int_{\epsilon}^{\infty} s^{-\theta} f\left(\frac{s}{S(\tau)}\right)\,\mathrm{d}s \qquad (4.3.8)$$

again, we apply the coordinate transformation: $x \equiv s/S(\tau)$ to change the variable of integration:

$$N(\tau) = (S(\tau))^{1-\theta} \int_{\frac{\epsilon}{S(\tau)}}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x \tag{4.3.9}$$

We can now see the central fallacy in the previous section's reasoning. The integral $\int_{\frac{\epsilon}{S(\tau)}}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x$ is not time independent because of the coordinate transformation acting on its limits. We know from the scaling ansatz that the function f is well behaved for small arguments. In fact, it had been assumed to be constant for arguments much smaller than 1. Here we will assume that f contains a "scaling correction":

$$f(x) \simeq x^{\delta} \tag{4.3.10}$$

for $x << 1$. Hence we may break the previous integral into a time dependent and a time independent component:

$$\int_{\frac{\epsilon}{S(\tau)}}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x = \int_{\frac{\epsilon}{S(\tau)}}^{\epsilon} x^{-\theta} f(x)\, \mathrm{d}x + \int_{\epsilon}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x \tag{4.3.11}$$

where we can approximate the time dependence as:

$$\int_{\frac{\epsilon}{S(\tau)}}^{\epsilon} x^{-\theta} f(x)\, \mathrm{d}x = \int_{\frac{\epsilon}{S(\tau)}}^{\epsilon} x^{-\theta+\delta}\, \mathrm{d}x = \frac{\epsilon^{\delta-\theta+1}}{\delta-\theta+1} - \frac{1}{(S(\tau))^{\delta-\theta+1}} \frac{\epsilon^{\delta-\theta+1}}{\delta-\theta+1} \tag{4.3.12}$$

We can use Eq. (4.3.12) in our expression for the normalization to find:

$$N(\tau) = (S(\tau))^{1-\theta} \left(\frac{\epsilon^{\delta-\theta+1}}{\delta-\theta+1} - \frac{1}{(S(\tau))^{\delta-\theta+1}} \frac{\epsilon^{\delta-\theta+1}}{\delta-\theta+1} + \int_{\epsilon}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x \right) \tag{4.3.13}$$

Hence giving us a time dependence for the normalization which shows a crossover:

$$N(\tau) = (S(\tau))^{1-\theta} \left(\frac{\epsilon^{\delta-\theta+1}}{\delta-\theta+1} + \int_{\epsilon}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x \right) - (S(\tau))^{-\delta} \frac{\epsilon^{\delta-\theta+1}}{\delta-\theta+1} \sim \tau^{-y} \tag{4.3.14}$$

If δ is small, in fact, if $\delta < \theta - 1$, the term $(S(\tau))^{-\delta}$ would dominate in the long term. Our simulations indicate that the correction to the scaling exponent is of the order of 0.1, which is significantly smaller than $\theta - 1$ and ergo, it seems plausible to argue that for a sufficiently large time, $(S(\tau))^{-\delta} \sim t^{-y}$. Hence: $\delta z = y$, proving Eq. (4.3.5).

The statement in Eq. (4.3.6) follows similar reasoning, except that for

$$\Phi\tau = (S(\tau))^{2-\theta} \left(\frac{\epsilon^{\delta-\theta+2}}{\delta-\theta+2} + \int_{\epsilon}^{\infty} x^{-\theta} f(x)\, \mathrm{d}x \right) - (S(\tau))^{-\delta} \frac{\epsilon^{\delta-\theta+2}}{\delta-\theta+2} \sim \tau \tag{4.3.15}$$

the $(S(\tau))^{2-\theta} \frac{\epsilon^{\delta-\theta+2}}{\delta-\theta+2}$ term dominates as $2 - \theta$ is positive. Hence proving Eq. (4.3.6).

Likewise, Eq. (4.3.7) follows the same reasoning, and too is independent of the time dependence of the integral limits in the long term. However, as our simulations indicate that the wetted area does not remain constant, we apply the ansatz that there exists a power-law time dependence for the wetted area. Thus:

$$W(\tau) = (S(\tau))^{\frac{d}{D}+1-\theta} \left(\frac{\epsilon^{\delta-\theta+\frac{d}{D}+1}}{\delta-\theta+\frac{d}{D}+1} + \int_{\epsilon}^{\infty} x^{-\theta} f(x) \, \mathrm{d}x \right) - (S(\tau))^{-\delta} \frac{\epsilon^{\delta-\theta+\frac{d}{D}+1}}{\delta-\theta+\frac{d}{D}+1} \sim \tau^{\sigma} \tag{4.3.16}$$

Which proves the result for Eq. (4.3.7). □

We may now use Theorem 4.1 to estimate the correction to the exponent of the power-law distribution θ_n. According to Theorem 4.1, the exponent of the size distribution ought to be: $\theta = 1 + \frac{d}{D} - \frac{y}{z} - \frac{\sigma}{z} = \frac{5}{3} - \frac{0.25}{2.87} - \frac{0.11}{2.87} \simeq \frac{5}{3} - 0.125$.

Considering that the exponent observed in our simulations is approximately $\frac{5}{3} - 0.104$, and that the error in fitting this exponent is difficult to determine, we may consider this theory a fair estimate for the correcting to the exponent of the size distribution.

4.4 Conclusion

We have reproduced the data of Family and Meakin, thus providing an important consistency check for our code. Furthermore, we were able to determine that the wetted fraction is possibly not constant in the system. If this is indeed the case, then one would expect important corrections to the theory put forward by Family and Meakin.

Furthermore, the traditional scaling theory did not include the fact that there exists a minimum droplet size and it was found that this finite lower bound does indeed constitute a correction to the scaling exponent of 5/3 being predicted by Family and Meakin's theory.

We were able to use additional information from our simulations to find a correction, or at least an upper bound, to the scaling exponent which agrees with our and Family and Meakin's numerical data.

Chapter 5

Hanging Droplets

> The purpose of computation is insight, not numbers.
>
> Richard Hamming

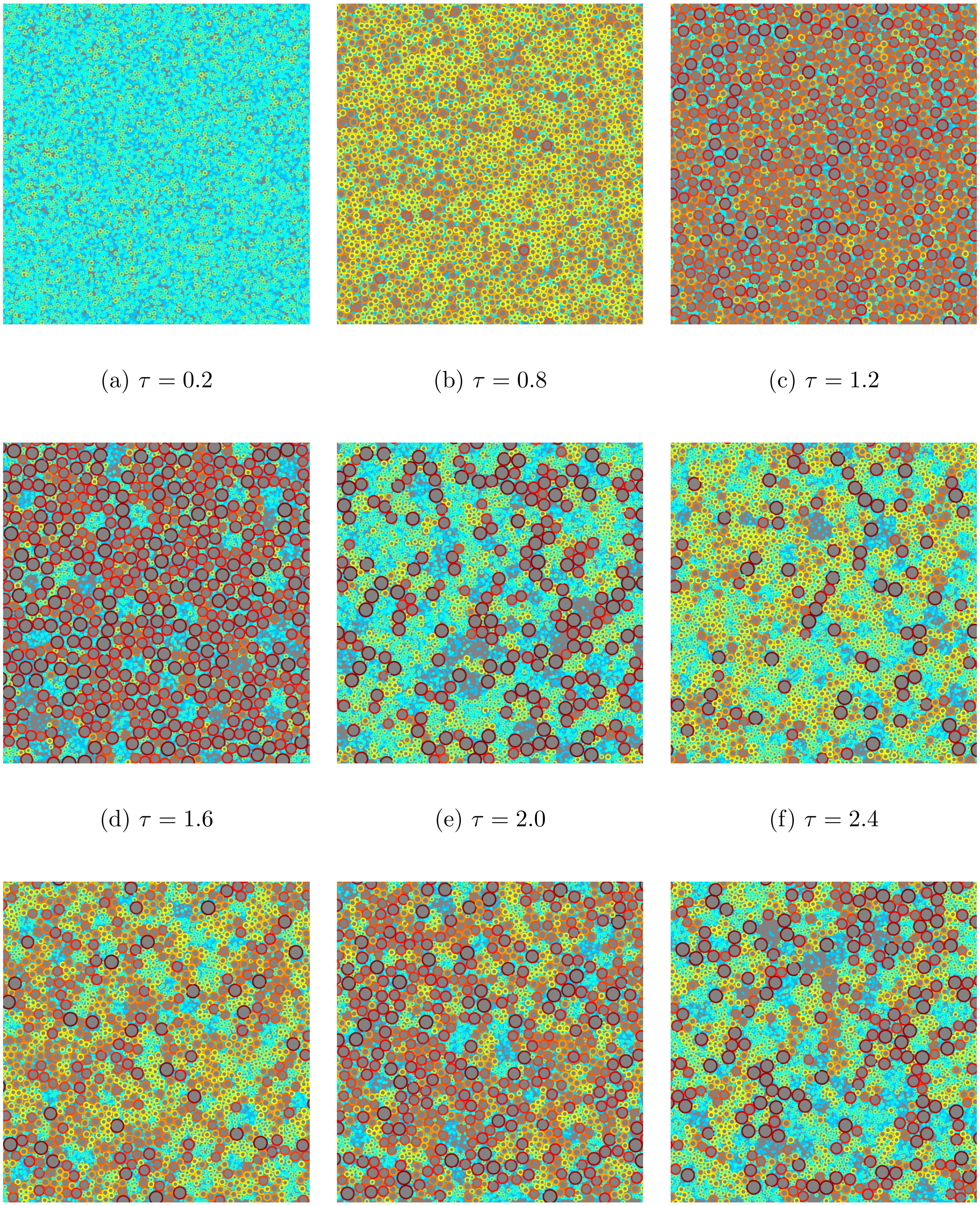
In this chapter we study the effect that gravity has on the evolution of breath figures. This is an interesting extension to the work of Family and Meakin, as their scaling theory no longer holds when droplets yield (let alone once they are distorted by gravity) because:

1. The scaling ansatz Eq. (1.2.1a) is not valid for yielding droplets as the system is no longer scale invariant near (and beyond) the yielding size.
2. The time dependence of the typical size Eq. (1.2.1b) disappears once droplets begin to yield (i.e., $z = 0$), thus leaving us with trivial form for Eq. (1.2.2b) (this essentially reads "a constant equals a constant").
3. The total mass in the system is no longer conserved, ergo we loose our time scale defined in Eq. (1.2.2a). Intuitively, when a yielding system is in the steady state, it is no longer possible to tell from the system's configuration how old it is.

These issues motivate us to extend the previous studies into the regimes of yielding droplets, and yielding droplets that have been distorted by gravity. In order to determine which property has what impact, yielding is first studied in the absence of droplet distortion by the gravitational field, and then the shape changes are introduced phenomenologically by employing Eq. (2.3.27).

An ansatz for a theory based on the steady-state master equation is also given.

The length and time scales have been defined in the same manner as in section 3.2.1. Likewise for the contact angle.



(a) $\tau = 0.2$

(b) $\tau = 0.8$

(c) $\tau = 1.2$

(d) $\tau = 1.6$

(e) $\tau = 2.0$

(f) $\tau = 2.4$

(g) $\tau = 2.8$

(h) $\tau = 3.2$

(i) $\tau = 4.0$

FIGURE 5.1: Snapshots of the simulations at different instants of time τ, for a system that is started empty at time $\tau = 0$. Colors encode the droplet radii.

5.1 Simulations with Yielding

On the way to simulating breath figure formation in gravity, we implement droplet yielding. That is, once droplets exceed a critical size, these droplets are reset to a size equal to the minimal droplet size.

5.1.1 Numerical Parameters

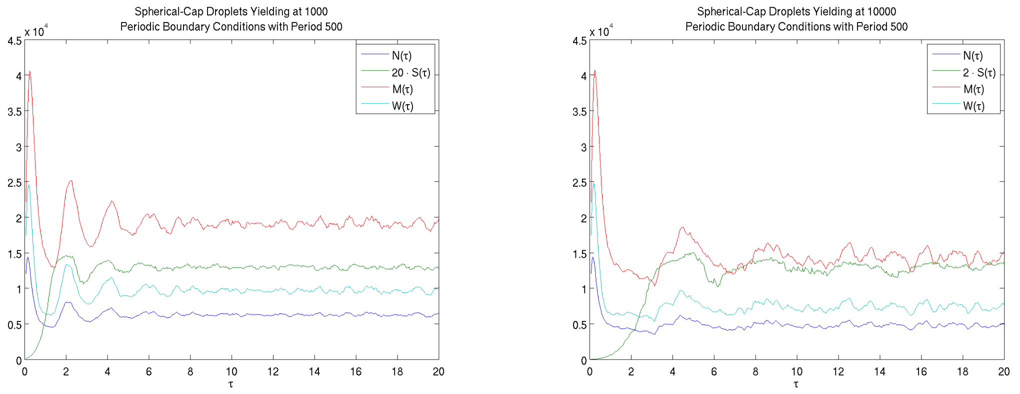
The system studied has periodic boundary conditions with a period of 500 and the largest (yielding) droplet sizes where 1000 and 10000. Two separate maximum sizes were simulated in order to determine the effect of setting an upper boundary for the droplet sizes. Also, for a maximum size of 1000 a system with periodicity 1000 was also simulated. After simulating for well over one month, a system with periodicity of $L = 1000$ and yielding size $s_{max} = 10000$ (which amounts to a yielding radius of $R = s_{max}^{1/3} \simeq 21.5 \ll L$ did not reach a steady state.

5.1.2 Pictures of System Evolution

Fig. 5.1 shows a selection of snapshots taken of the simulation in order to provide an overview of the evolution of breath figures as simulated by our model.

When the growth of a breath figure begins and the system is sparsely occupied, then the likelyhood of a new droplet overlapping with an existing one is very small. Thus the system is initially populated with droplets of almost all the same size (Fig. 5.1(a)). Once system is densely packed, the likelyhood of a new droplet landing on unoccupied space is very small. And since all droplets have almost the same size, the probability of being struck by a new droplet is effectively equal for all droplets in the system. This leads an almost mono-disperse population (except for some lucky small droplets which have found a gap) which is evenly growing (Fig. 5.1(b)). Since the system is tightly packed, coalescence would become the dominant contributor to growth [3]. This continues until most droplets are close to the yielding size (Fig. 5.1(e)). At this point a large quantity of these large droplets drip off the substrate. This leads to the oscillations in droplet number, characteristic size, wetted area, etc as clearly seen in Fig. 5.2. Over time, polydispersity develops as these oscillations become decoherent and global oscillations can no longer be observed (Fig. 5.1(i)). However, on a local scale, droplets still prefer to be adjacent to droplets of similar size.

We may estimate the time for which the first droplets begin to yield in the following way: Consider a system where most droplets have a size equal to the yielding size s_{max}. Thus each droplet has an area of $\pi r_{max}^2 = \pi s_{max}^{2/3}$. The time scale of τ is essentially a



(a) Yielding size 1000

(b) Yielding size 10000

FIGURE 5.2: The number of droplets $N(\tau)$ (blue line), the characteristic size $S(\tau)$ (green line), the total mass $M(\tau)$ (red line) and the wetted area $W(\tau)$ for yielding sizes of (a) 1000 and (b) 10000. In both cases, these quantities exhibit some temporal oscillations of the same phase and approximate frequency and then become coherent, eventually settling at a constant value.

measure of how many droplets of minimal size s_{min} had been deposited onto every part of the system. If a droplet yields because it coalesces with a neighbor of equal size, then number of minimal droplets involved is approximately $2A\tau^1$ where A is the area of the maximal droplet. Thus, in a system populated by droplets of equal and maximal size, the first yielding events will occur when:

$$2\tau A \simeq s_{max} \Rightarrow \tau_{yield} = \frac{1}{2}\frac{s_{max}}{A} = \frac{1}{2}\frac{s_{max}}{\pi s_{max}^{2/3}} \tag{5.1.1}$$

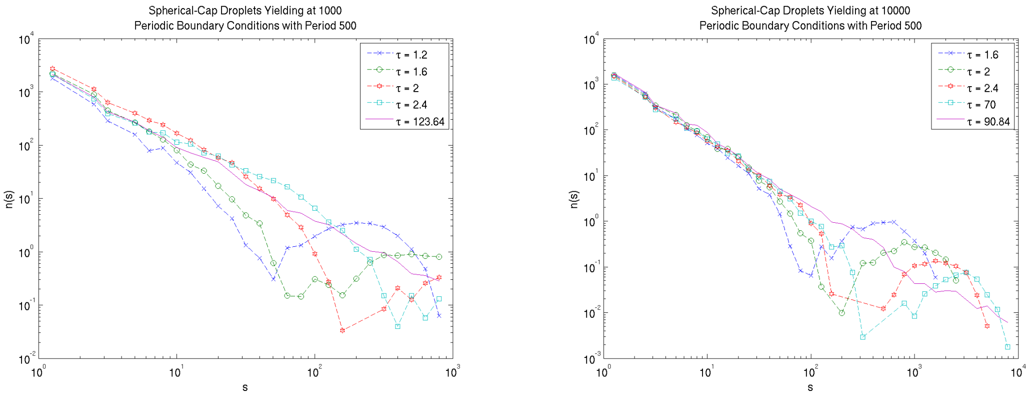
for $s_{max} = 10^3$ we get $\tau_{yield} \simeq 1.59$ which is in agreement with what is observed in Fig. 5.1.

The frequencies of the oscillations differ by a factor of approximately 2 when going from a yielding size of 10^3 to a yielding size of 10^4. This can be understood by assuming that locally the system exhibits the same characteristic size $S(\tau)$ (Eq. (1.2.1b)) as with the case of sessile droplets^2. The we may estimate the difference in the periods is by asking: how much longer do we need to wait for the droplets to reach the yielding size when we increase the system size by a factor of 10? Hence, if the yielding sizes differ by a factor of 10:

$$10 = \frac{S(\tau_1)}{S(\tau_2)} = \left(\frac{\tau_1}{\tau_2}\right)^z \tag{5.1.2}$$

[1] The factor of 2 comes from the fact that when examining Fig. 5.1(d) we can see that most droplets have a size close to the yielding size, yet for coalescence to cause yielding, two such droplets will be involved.

[2] While this might not be the case globally, we assume that locally, droplets in a system with a yielding size, still behave approximately as sessile ones.



(a) Yielding size 1000

(b) Yielding size 10000

FIGURE 5.3: The size distribution $n(s)$ of spherical cap droplets for systems with 2 different yielding sizes. The period is 500 and 1000 respectively

hence, using our simulated $z_{sim} = 2.88$, we predict that $\tau_1 \simeq 2.2\tau_2$. Indicating that the oscillations in the system mass M are due to local patches yielding in unison.

5.1.3 Size Distribution

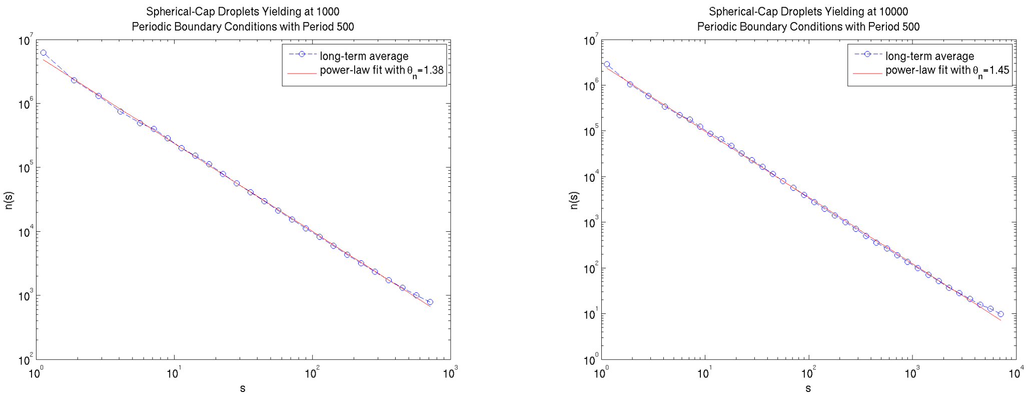
Fig. 5.3 shows the evolution of the number of droplets $n(s)$ of size s observed in our system for yielding sizes for 1000 and 10000 respectively.

All averages over time where taken after the oscillations shown in Fig. 5.2 have died outin order to ensure averages over stationary and statistically independent samples.

As visible in Fig. 5.3, droplet yielding erases the bump once it "runs into" the yielding size. Contrary to the system without yielding, we may now average the size distributions over time τ once no further correlations are apparent in the system. The coherence time may be read from Fig. 5.2 in order to ensure that no fluctuations affect the result of the averaged size distribution.

Fig. 5.4 shows the average over the long term of the size distribution. When yielding is included, the exponent θ_n changes from 1.56 to 1.38 for yielding at 1000 and to 1.45 for yielding at 10000.

We have not yet come to an explanation regarding this difference in exponent. One might suspect that this is an artifact of the fitting algorithm and that the distributions are actually the same. This may be easily tested by plotting the reduced plots as seen in Fig. 5.5. Here we have divided the data by the fitted power law for exponent 1.45. It is interesting to note that both for both system sizes, the reduced plots show the same characteristic shape. This is indicative of corrections to the power-law distribution as



(a) Yielding size 1000

(b) Yielding size 10000

FIGURE 5.4: The effect of different yielding sizes on the size distribution of droplets. The exponent of the size distribution changes from 1.38 to 1.45 when the yielding size increases from 1000 to 10000. This is not an artifact due to a finite system size as this change persists when conducting simulations for periodic boundary conditions with periodicity 1000 as apparent from Fig. 5.6

statistical noise should have produced a random pattern.

At this point, one might consider that finite size effects are responsible for the much smaller exponent compared to the non-yielding result. This was tested in the most direct manner by simulating a system with yielding 1000 and periodicity 1000. As we can see from Fig. 5.6, increasing the system size by a factor of 4 did very little to change the exponent.

Hence we may conclude that systems with yielding have a significantly different size distribution, in particular, the size distribution remains a power law (without a "bump") and that the exponent of the power law is dependent on the yielding size. This indicates that the droplet size distribution is affected as a whole (and not just the distribution for large droplet sizes) if very large droplets are removed from the system.

It is unclear whether the exponent should approach that of no yielding for increasing yielding size, as the distributions for yielding and no yielding are fundamentally different.

5.1.4 What about the "Bump"?

A question left open by FAMILY AND MEAKIN is the peculiar "bump" in the size distribution around the biggest droplet sizes. This indicates that there is an abnormal number of large droplets.

An explanation of this may be found by analyzing the age of droplets. Each droplet is

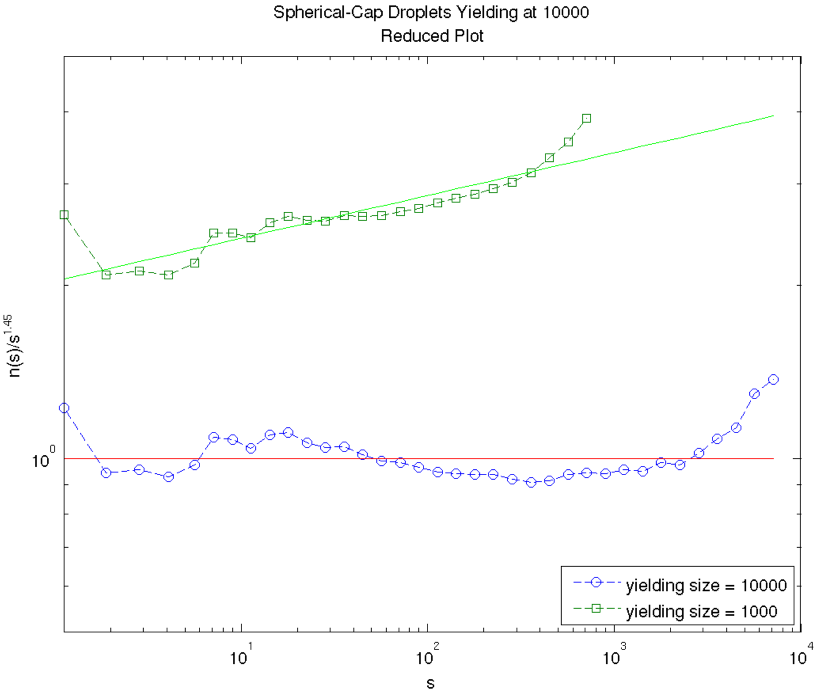


FIGURE 5.5: Examining the size distributions for 2 values of yielding: 1000 (green) and 10000 (blue). We have plotted the reduced plots in order to determine a systematic deviation in the distribution for yielding size 1000 from the distribution for yielding size 10000. One can clearly see a real difference in the distribution's exponent. Hence our findings in Fig. 5.4 are not an artifact of the fitting method.

assigned a "date of birth" upon creation as described in section 3.2. Yielding is expected to cause some kind of erasure of the initial state of the system.

Fig. 5.7 shows that before the first generation of droplets fall from the substrate most droplets in the system are about as old as the system itself. This is nothing more than a re-hash of what we can see qualitatively in Fig. 5.1; before the droplets are old enough to fall off the substrate, the first generation dominates the system. However, once yielding sets in, the age distribution changes from being dominated by the oldest droplets, to being exponentially distributed. Hence there is no longer a "characteristic age" in the system.

As this occurs at approximately the same time as the "bump" disappearing, we are lead to conclude that the "bump" in the droplet size distribution does indeed correspond to the oldest droplets in our system. These would persist in systems without yielding, consequently, the "bump" is caused by the initial state persisting throughout the evolution of the breath figure.

However, once yielding is included in the evolution of the breath figures, the initial homogeneous configuration is ultimately erased, leaving a memory-less system and hence

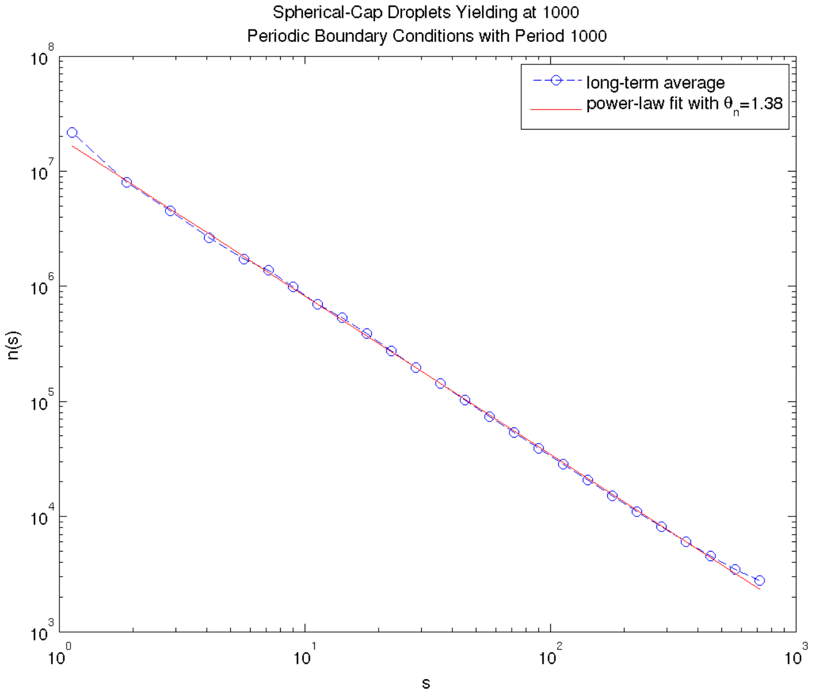


FIGURE 5.6: In order to check the result in Fig. 5.5, we plot the size distribution for a system with yielding at 1000 but with double periodicity. The exponent of the distribution is effectively the same as Fig. 5.4, yet the droplet number is approximately 4 times greater. This is expected as the system is also 4 times greater.

no characteristic sizes, nor ages of droplets.

This is also in agreement with Fig. 5.2, as the bump disappears only shortly before the oscillations in droplet number, characteristic size, total mass and wetted areas begin to die out, indicating that the bump is related to a temporal correlation of droplet distributions.

5.2 Simulations with Gravity

In the previous section, we have dealt with gravity by simply removing droplets from our system once they reach a critical yielding size.

As we have seen in chapter 2, the profiles of hanging droplets correspond to solutions of a coupled system of ordinary differential equations which needs to be integrated every time the volume of a droplet changes. This is prove to be a computationally very intensive exercise if every coalescence event was to result in integrating said PDEs, followed by iteration to fix the contact angle.

Instead, Eq. (2.3.27) presents itself as it has a closed form for writing but the volume

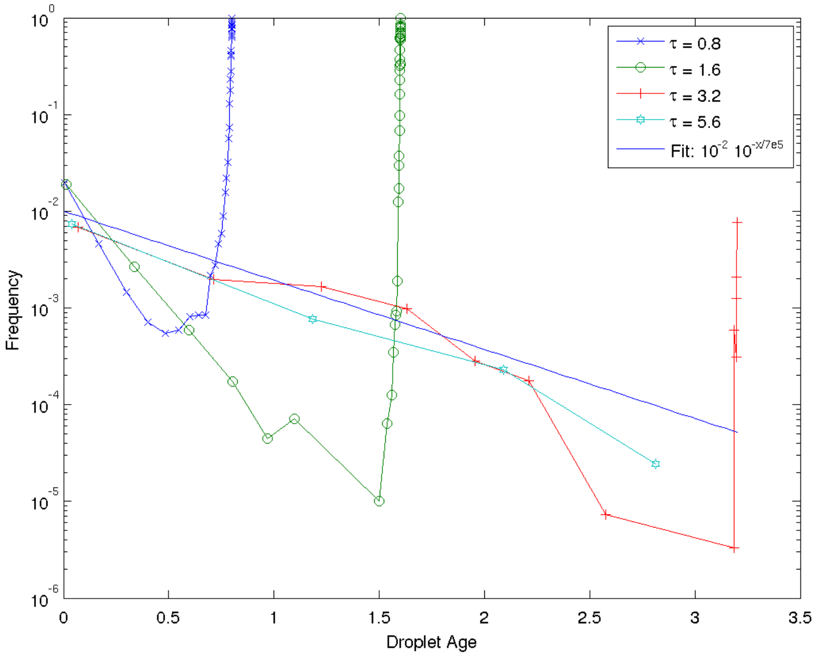


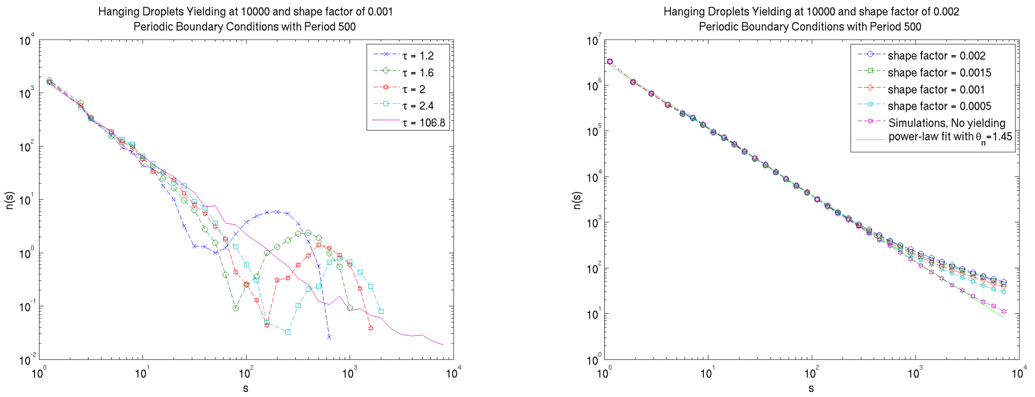
FIGURE 5.7: The distribution of droplet ages. Yielding onset is between 80 and 140 time steps. Before droplets are big enough to yield, most droplets have a very similar age with only a minority of droplets being very "young" and the rest aging at approximately the same rate as the system. This indicates that these droplets were indeed those from the initial "wave" seen in Fig. 5.1. When yielding sets in, the oldest droplets are removed, ultimately erasing the age distribution's bias for very old droplets. In the steady-state, the droplet ages are distributed according to an exponential distribution – as expected for random collisions.

of a droplet as a function of radius and radius as a function of volume.

Hence in this section we have used the relation $s(r) = r^3 + \hat{\kappa} r^5$ (Eq. (2.3.27)) for the volume of droplets instead of the $s(r) = r^3$ which was used in the previous section.

The idea is fairly simple: for the shape factor, $\hat{\kappa}$, very small[3], small droplets will appear to be effectively the same as spherical caps. However, their shape will deviate significantly from spherical caps for large droplets.

[3] The shape factor in section 2 refers to systems where the size scale is set by the capillary length λ_c. In these numerical systems, the length scale is fixed by the maximum droplet size 10^4. In order to make these numerical simulations comparable to the calculations from section 2, we choose our contact angle $\alpha = 59°$, corresponding to a yielding radius of approximately 1.5 capillary lengths. Hence we have an equivalent capillary length of $\lambda_{sim} = \left(\frac{10^4}{1.4}\right)^{1/3}$ It can then be shown that in these units, $\hat{\kappa} = \frac{\kappa}{\lambda_{sim}^2}$ where κ is defined in Eq. (2.3.27)



(a)

(b)

Figure 5.8: The size distribution of droplets hanging under the influence of gravity. The shape parameter simulated was $\hat{\kappa} = 1 \cdot 10^{-3}$. The distribution for the steady state clearly shows a crossover from one power-law distribution to another for sizes around 10^3. For $s \ll 10^3$, the exponent is $\theta_n = 1.45$ which is extremely close to the exponent of the distribution for yielding droplets which remain spherical caps (green line). The exponent for $s \gg 10^3$ is $\theta_g = 0.81$. Fig. 5.8(b) shows this for a selection of different shape factors, $\hat{\kappa}$. Changing $\hat{\kappa}$ appears to move the crossover.

5.2.1 Numerical Parameters

As with section 5.1, the system studied has periodic boundary conditions, with a period of $L = 500$, and the largest droplet sizes where $S = 10000$.

In keeping with Eq. (2.3.27), we simulated systems with $\hat{\kappa}$ ranging from $5 \cdot 10^{-4}$ to $2 \cdot 10^{-3}$ in steps of $5 \cdot 10^{-4}$. This corresponds to a significant deviation from the spherical cap behavior for sizes $s \gtrsim 10^3$.

5.2.2 Results

Fig. 5.8 shows the size distribution if the shape factor, $\hat{\kappa}$ was chosen to be 0.001. We can see that the steady-state distribution has 2 clear power-law regimes with a fairly sharp cross-over. This crossover occurs at approximately 10^3. This is sensible, for r^3 dominates $\hat{\kappa} r^5$ in Eq. (2.3.27) for values of $s = r^3$ smaller than κ. The 5th order guarantees that the crossover is fairly sharp.

We can see from Fig. 5.8 that the distribution is largely unaffected for sizes much smaller than the crossover. In fact, the exponent of 1.45 is statistically indistinguishable from the exponent of 1.45 for the case of spherical droplets yielding at 10000. This indicates that droplet's deviating from spherical caps does not affect the distribution of the population much smaller than the crossover size. However, once droplets reach a critical size, their distribution flattens out, still being power-law distributed but with an exponent of 0.81. That the distribution ought to flatten out once droplets deviate from the spherical cap shape is sensible, as droplets become deeper. This means that a deep droplet may have a larger volume for a given wetted radius than a shallow droplet, and so more larger droplets may “fit” onto the substrate.

Since scaling arguments rely entirely on there not being a yielding size, we cannot offer a theoretical means for determining the exponent for non-spherical droplets.

However, Fig. 5.8(b) inspires us to conjecture that there exists a universal distribution onto which all distributions with a specific crossover may be scaled. We may estimate the crossover length as follows:

$$s(r) = r^3 + \hat{\kappa} r^5 = r^3 \left(1 + \hat{\kappa} r^2\right) \tag{5.2.3}$$

hence we expect deviation from the spherical cap shape once the quantity in the brackets starts to significantly deviate from 1. Hence we may estimate the crossover length, r_c, to be:

$$\hat{\kappa} r_c^2 = 1 \Rightarrow r_c = \frac{1}{\sqrt{\hat{\kappa}}} \tag{5.2.4}$$

Ergo we postulate that the steady-state size distribution of droplets that are distorted by gravity flows:

$$n(s, r_c) = s\left(r_c\right)^{\theta_n} n_c \left(\frac{s}{s\left(r_c\right)}\right) \tag{5.2.5}$$

where θ_n is the exponent of the size distribution for yielding by not distorted droplets and n_c is a universal function.

Interestingly though, we found this not to be the case. Instead the rather peculiar scaling shown in Fig. 5.9 was found. Hence, once droplets are distorted by gravity, (as modeled by Eq. (2.3.27)), the size distribution follows the scaling:

$$n(s, r_c) = r_c^{\theta_n} n_c \left(\frac{s}{r_c}\right) \tag{5.2.6}$$

with the universal function n_c having form:

$$n_c\left(\frac{s}{r_c}\right) = \begin{cases} \left(\frac{s}{r_c}\right)^{-\theta_n} & \text{for } s < 20 r_c \\ \left(\frac{s}{r_c}\right)^{-\theta_g} & \text{for } s > 20 r_c \end{cases} \tag{5.2.7}$$

Where $\theta_n = 1.45$ is the exponent for the case of just yielding and $\theta_g = 0.81$ is a measured exponent possibly depending on yielding size but independent of r_c. Eq. (5.2.7) is plotted in Fig. 5.9 by rescaling the curves from Fig. 5.8(b). We cannot offer an explanation for the strange units of s/r_c or why the crossover takes place when $s/r_c \sim 20$.

We find that altering the shape factor, $\hat{\kappa}$ does not appear to significantly alter the exponent of the distribution for either regime.

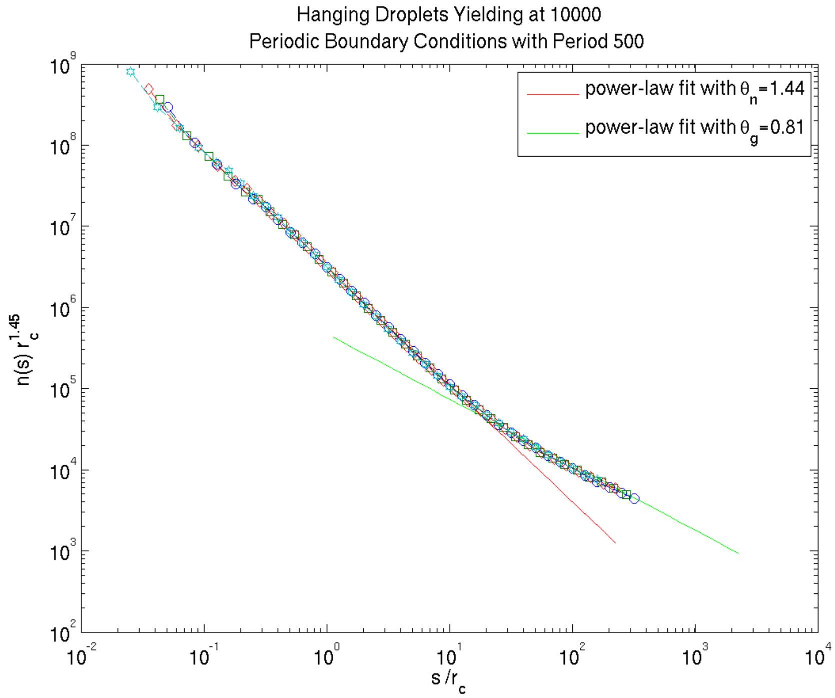


FIGURE 5.9: The distribution of droplets for different shape factors ($\hat{\kappa}$) having been rescaled. These appear to all lie on one universal curve.

5.3 Master Equation Arguments

As detailed at the beginning of this chapter, scaling theories of the same kind as the one proposed by FAMILY AND MEAKIN are incompatible with systems that have some form of yielding. This is essentially due to the scale invariance breaking caused by an upper-size, that does not scale with time. Ergo, the time and size scales are no longer related simply by rescaling.

However, since the system evolved by a simple surface contact dynamics, we may make the assumption that droplets largely interact via pairwise mergers, and thus the dynamics of our system will be given by the master equation:

$$\frac{\mathrm{d}n(s,t)}{\mathrm{d}t} = \sum_{k=1}^{s-1} \Pi(s-k,k,t) - \sum_{k=1}^{\infty} \Pi(s,k,t) \tag{5.3.8}$$

where $\Pi(s,k,t)$ is the likelyhood that 2 droplets with sizes s and k with merge at time t to give a droplet of size $s+k$.

We may write the master equation in terms of the conditional probabilities $K(s,k)$, the probability that 2 droplets of size s and size k merge, given that these droplets exist in the system.

$$\frac{\mathrm{d}n(s,t)}{\mathrm{d}t} = \sum_{k=1}^{s-1} K(s-k,k)n(s-k,t)n(k,t) - \sum_{k=1}^{\infty} K(s,k)n(s,t)n(k,t) \tag{5.3.9}$$

where the assumption that K is independent of time has been made. This assumption is not unreasonable, and has also been proposed by FAMILY AND MEAKIN.

Consider now the steady state, $\frac{\mathrm{d}n(s,t)}{\mathrm{d}t} = 0$. Its distribution $n(s)$ satisfies:

$$0 = \sum_{k=1}^{s-1} K(s-k,k)n(s-k)n(k) - \sum_{k=1}^{\infty} K(s,k)n(s)n(k) \tag{5.3.10}$$

which may be simplified to:

$$n(s)F[n](s) = \sum_{k=1}^{s-1} K(s-k,k)n(s-k)n(k) \tag{5.3.11}$$

with the functional F defined as $F[n](s) \equiv \sum_{k=1}^{\infty} K(s,k)n(k)$. This equation is of quite the intuitive form: the number of droplets that have size s ought to be a linear combination of all droplets whose sizes add up to s. This is because a droplet of size s could have formed from a droplet of size $s-1$ and 1, or $s-2$ and 2, and so on. The intuition is that this combinatoric form is described by Catalan numbers (see Fig. 5.10) which can have a power-law distribution (depending on initial conditions) with exponent 1.5. These numbers are well known in combinatorics and form the starting point of this investigation.

We shall provide only a beginnings of a theory here, to at least proof that the steady-state master equation *can* yield a power-law distribution. This is by no means intended to be a complete picture. Such an analysis falls within the future scope of this research.[4]

5.3.1 Trivial Correlation

If the functional $F[n](s)$ has the trivial form $F[n](s) = s^{2\lambda}$ with $K(s,k) = (sk)^{\lambda}$, then Eq. (5.3.11) may be written as:

$$\mu(s) = \sum_{k=1}^{s-1} \mu(s-k)\mu(k) \tag{5.3.12}$$

where $\mu(s) \equiv s^{\lambda}n(s)$

Then the Eq. (5.3.12) has the solution according to VARDI [15]:

$$\mu(s) = \mu(1)^s \frac{1}{n+1}\binom{2n}{n} \simeq \frac{(4\mu(1))^s}{\sqrt{\pi}s^{3/2}} \tag{5.3.13}$$

[4] Since it could not be completed in time for the submission of this thesis

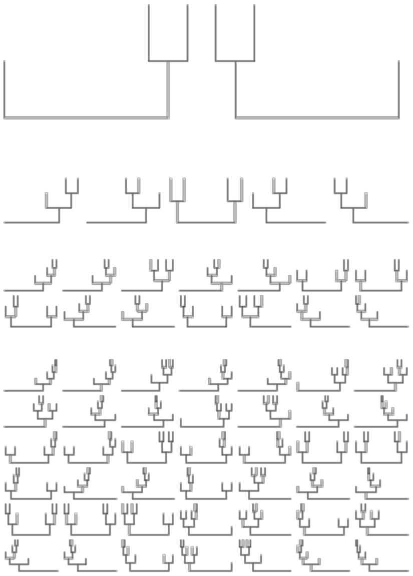


Figure 5.10: Demonstrating the philosophy of Catalan numbers and their connection with the combinatorics of droplet coalescence. The topmost binary trees represent the combinations 3 droplets may merge to form a single droplet of size 3, there are 2 possible ways. The second (from the top) shows 4 droplets merging to give a droplet of size 4, there are 5 combinations. The next set represent 5 droplets merging, there are 14 possible combinations. Finally the lowest set shows the 42 combinations 6 droplets can merge using only pairwise mergers.

the final expression containing the power law is the asymptotic limit which is reached quickly. Consequently, for such correlations, one may observe exponential tails for the initial condition $\mu(1) \neq 1/4$. It is of course entirely unrealistic to have divergent distributions for Eq. (5.3.13), and thus we may restrict the initial condition $\mu(1) \leq 1/4$. In the case of $\mu(1) < 1/4$ Eq. (5.3.13) takes the form of a power law with exponent -1.5 but with an exponential cutoff.

Hence, depending on the trivial correlation function's exponent λ, this theory predicts a power-law exponent for the steady state size distribution of $-1.5 + \lambda$.

5.3.2 Expected Correlations

Having examined the solution to Eq. (5.3.11) for a conceivable correlation function, we might ask ourselves, if there are no preferential pairs of droplet sizes in our system, what would be the "geometric" mean field correlation? In other words, if we placed random droplets onto our substrate what droplets ought to be adjacent merely from considering how many droplets fit within their vicinity such that after a short time interval they would overlap.

Fig. 5.11 shows one such possible state: two adjacent droplets of radii r and r' initially

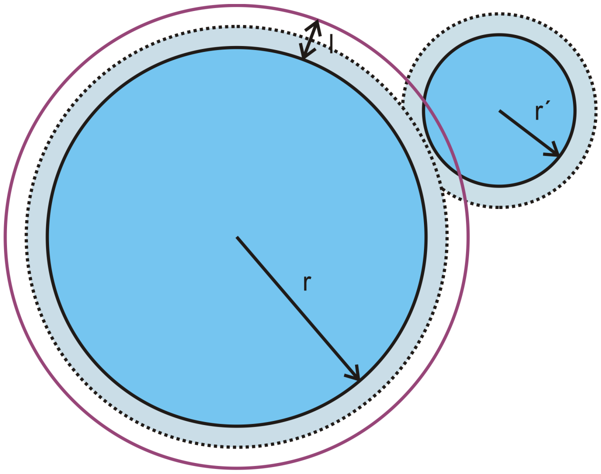


FIGURE 5.11: A prediction for the correlation function for the geometric mean field: i.e., the number of droplets of radius r' that can fit beside a droplet of radius r such that it is in a position to merge. As an approximation: both droplets might grow in radius by about $l/2$ per short time interval. Thus we want to determine the probability of finding a droplet in such a position that it does not yet overlap, however it will overlap after a very short time. This would constitute the droplets centers of mass falling within a distance of $r' + r$ to $r' + r + l$.

do not touch. Making the approximation that both droplets grow in radius by about $l/2$ for a given short time interval, then the droplet's centers of mass must be closer than $r' + r + l$ to coalesce within this short time interval. Yet, they must also be further apart than $r' + r$ otherwise they would already overlap. This means, if we hold the droplet of radius r fixed, then a droplet of radius r' must have its center of mass within a strip of area approximately $2\pi l \left(r' + r + \frac{1}{2}\right)$.

Hence the likelyhood that two droplets merge, $\Pi(s, k, t)$, can be approximated provided no preferential "pairing up":

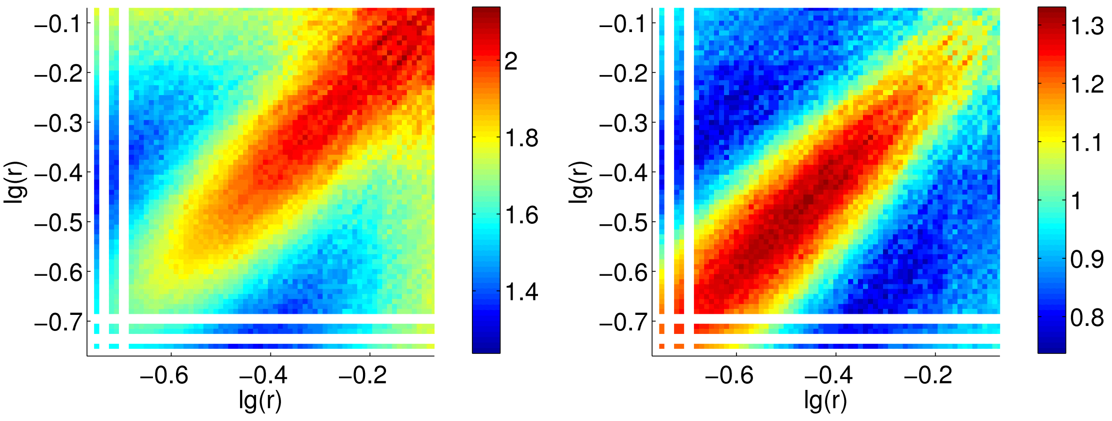
$$K_{mft}(s, k) \approx 2\pi l \left(s^{\frac{2}{3}} + k^{\frac{2}{3}} + \frac{l}{2} \right) \tag{5.3.14}$$

Unfortunately, this is nothing like the "trivial" pair correlation needed to solve the problem exactly. Instead Eq. (5.3.11) needs to be solved numerically. These numerical solutions, and their comparison to the simulated data will be the focus of future work.

5.3.3 Correlations from Simulations

Fig. 5.12 shows the pair correlation of the simulated data. For these graphs, $l = 0.25$ (as defined in Eq. (5.3.14)). These graphs were generated by the LAPP's code used to also analyze the experimental data.

We can clearly see that the simulations agree with the assessment that the correlation function K, is not of the trivial nature that is necessary for the analytical solution



(a) $K(s, k)$ measured from the simulated data

(b) The deviation from the geometric mean field theory: $K(s, k)/K_{mft}(s, k)$

FIGURE 5.12: The conditional probability function K "measured" from simulated data using the analysis program developed by LAPP for the experimental data. The axes show the logarithm of droplet radii normalized to the yielding size. The color scale represents the conditional probability K, the values on the scale have not been normalized for droplet number. Fig. 5.12(a) shows K as measured by considering all neighbors for $l = 0.25$ (in units of the smallest droplet size as defined in Eq. (5.3.14)). Fig. 5.12(b) shows K as measured from the simulated data divided by the mean field theory (Eq. (5.3.14)). Here the deviation from the mean field theory defined in Eq. (5.3.14) is clear: droplets have preferential neighbors of similar size. The white stripes are caused by the discrete droplet sizes.

derived in section 5.3.1. Droplet sizes in our simulated breath figures appear to be correlated such that droplets of similar sizes are more likely to be adjacent to each other than droplets of different sizes.

While there is no conclusive evidence for what exactly causes this kind of correlation in neighbor size, we can offer an argument which demonstrates that the observed correlations are consistent with the existence of local domains in which all droplets had nucleated at approximately the same time.

When area on the substrate is freed due to coalescence and yielding, it is usually very quickly filled up with new droplets. When this area is freed due to yielding, it is usually very large and is ultimately seeded with droplets that initially do not touch. This can be observed in Fig. 5.1 and from the different droplet growth rates (pre-merging and merging) reported by BEYSENS *et al* [3].

Hence it is reasonable to assume that the breath figure has local regions of droplets of a similar age (and thus size) which have nucleated in the same area. For droplets in these local regions, their most immediate neighbors would be of a similar size. Even though this effect would not be present at the edge of such a region, such local inhomogeneity can have an effect on the correlation function if these regions are large enough. Since droplets in such a region would also yield at approximately the same time, these regions could persist in the steady state.

Alternatively this observed preferential size for neighboring droplets can be thought of as indicating that local regions with a characteristic size (similarly to the bump the Family and Meakin theory) exist in steady state.

Running through the full set of frames, some of which are displayed as Fig. 5.1, show local regions nucleating, coarsening and then yielding.

Exponent of Size Distribution in Gravity

Assuming trivial correlations, one ought to expect an exponent of $-1.5 - \lambda$ for the size distribution for droplets that do not change their shape and yield at some maximal size. However, it is clear from this chapter that the exponent for yielding droplets is much smaller. The fact that the theory assuming trivial correlations does not agree with the observed exponents is to be expected as our study of the correlation revealed a non-trivial correlation far from the mean-field.

5.4 Conclusion

We have studied the effect of gravity on the evolution of breath figures in the context of droplets hanging from a horizontal planar surface with periodic boundary conditions.

To this end we have introduced a yielding mechanism which removes droplets from the substrate once they reach a particular critical size. In this work the effect of 2 maximum droplet sizes was studied, 1000 and 10000 (in units of smallest droplet size).

We have found that there are large differences in the droplet size distribution when compared to the case of sessile droplets that are not distorted by gravity. There is no "bump" present in the steady state. Rather we observe a pure power law over all droplet sizes. Furthermore, the exponent of this power law is very different for the case of yielding droplets than in the case of sessile droplets where the exponent is also dependent on the yielding size. For a yielding size of 1000, the exponent was measured to be 1.38 and for a yielding size of 10000 the exponent was measured to be 1.45.

Furthermore, we used our work on the shape of hanging droplets to determine the effect of gravitational distortion on the droplet size distribution. We have found that shape distortions introduce a novel crossover from one particular power law (with the same exponent than the case of just yielding) to a power law with exponent 0.81 for droplets greater than a specific crossover size. Furthermore, there exists a universal distribution which is simply rescaled by the crossover length when the magnitude of the gravitational distortion is changed (this corresponds to changing the contact angle or the magnitude of the gravitational forces).

In order to better understand this system, a simple theory based on a steady state master equation was formulated. For non-correlated droplets this theory predicts an exponent of 1.5 which is close for the system with a large yielding size. In order to understand this theory better, the pair correlation function was measured from our data. We found that this correlation is not trivial and deviates from the mean field. While not being explored any further, it is our suspicion that this correlation is responsible for the deviation in the exponent of the size distribution. We have proposed arguments based on the similar age of neighboring droplets in order to account for this non-mean field correlation of droplets.

Chapter 6

Conclusion

> Still round the corner
> there may wait
> A new road or a secret gate.
>
> J. R. R. Tolkien, *Lord of the Rings*

Since the theory available at the beginning of this study could not give satisfactory answers to questions posed by recent experiments of LAPP, sophisticated code was constructed in order to advance theory.

Using this, we where able to determine that the bump observed close to the largest droplet sizes is caused by neighboring droplets having a very similar age. Hence droplet ages are very important, and the initial state of the system persists for a long time. Yielding causes this state to eventually be deleted and consequently the bump disappears.

Furthermore, we have been able to show that the minimum size of droplets due to nucleation (or rain) is important and has consequences on the exponent of the size distribution. This has allowed us to explain the deviation from the theoretical value of 5/3 for the power-law exponent which was observed by FAMILY AND MEAKIN and reproduced by our simulations.

Investigations into the effect of gravity on the distribution of hanging droplet sizes yielded two important observations: 1) the yielding size of droplets has an effect on the exponent of the size distribution, with the larger yielding size producing a larger exponent, and 2) shape distortion of the droplets results in a novel crossover from one power law to another. This scales onto a universal distribution consisting of two power-law regimes. The regime for small droplets is unchanged compared to the system where we have just yielding. The regime of droplets larger than the crossover length has a universal exponent of 0.81 irrespective of crossover length.

Outlook

We have found that breath figures are a very fruitful system to examine. There are many stones that we have had to leave unturned due to time constraints. Here we will give a short outline of what is on the horizon.

Droplet Selection Algorithm

After having implemented the algorithm discussed in chapter 3, a potential improvement of speed came to mind. When selecting a voronoi cell which to add a droplet too, we go through the list of all droplets until the random number which we picked is less than the sum of the areas of the preceding voronoi cells. Initially one might think that this algorithm ought to scale as $O(N/2)$ with N being the number of droplets. This is not entirely true. The list of droplets is appended to by new droplets, and so the youngest (and thus smallest droplets) are at the back resulting in the droplets which are oldest and thus are most likely to have the largest voronoi cell areas being tested first. And so this algorithm is faster than $O(N/2)$.

However, this problem has been readily solved by using binary tree searching algorithms instead: we ought to construct a binary tree and fill it completely with the droplets in the system. Then at each node with place the sum of the voronoi cell areas of its children. Then we may select a voronoi cell by using a binary tree search which scales as $O(\log n)$. Of course this is far less trivial as described here because all changes in the triangulation would result in updates to each node and the tree structure would change (potentially considerably) whenever droplets are deleted or merge. This has prevented us from implementing this improved algorithm already.

However new evidence has come to light that this might be the slowest step for extremely large systems, motivating us to redoubling our efforts to implement this change.

Shape of Sessile Droplets

As yielding provided us with an interesting theoretical question, we had focused our attention on the shape of hanging droplets. However, in principle Wente's equations can also be applied to sessile droplets in gravity (with some small modifications). Hence, our method for fixing the contact angle should also be applicable there.

Master Equation Theory for Hanging Droplets

In order for us to understand the data for hanging droplets, we put forwards a master equation approach which copes with yielding and shape distortion. So far, a theory for yielding droplets was started. Which already provides a faint glimpse into what such a theory might look like.

We were able to commence the analysis of the correlations of droplet sizes (and their ages). In the near future, we will complete our analysis of the correlation of droplets by examining which droplets actually do merge (rather then merely being very close together).

Furthermore we will devote significantly more time to solving the master equation for the case of non-trivial correlations.

Modifications to the Scaling Theory for Sessile Droplets

It is not clear from our data, whether the evolution of breath figures will ultimately yield a system whose wetted area is constant with time. However, this does seem unlikely since having a lower bound for the droplet size ought to also change the wetted area under rescaling (since rescaling effectively reduces this lower bound, allowing smaller voids to be filled and this increasing the wetted area). Our simulations have not been running for long enough in order to verify this hypothesis. Simulations of bigger systems are under way and should reach longer times in some weeks time. The analysis of this data is one of our most immediate goals once said data becomes available. This would allow us to check the asymptotic behavior of sessile breath figures and consequently would shed some light onto the extremely long term behavior.

The formation and evolution of breath figures turns out to be a fruitful subject to analyze. The present thesis describes advances in the area of the growth of breath figures on the underside of a substrate. From this investigation, it is clear that the distribution of droplet sizes changes considerably when introducing gravitational effects such as yielding and droplet shapes that are no longer spherical caps. Yet, we have only glimpsed the full extend of these phenomena. We envision for the future that the development of a theory to be based on the pair correlation function will produce many more and interesting extensions to this work.

Chapter 7

Acknowledgments

Firstly, I would like to thank Jürgen Vollmer of the Max Planck Institute for Dynamics and Self-Organization for his guidance, support and indispensable help on this project. He had first suggested that the existing theories on breath figures could be extended, and thus ultimately inspired me to look closely at this everyday phenomenon. Jürgen had succeeded in inspiring me that I do not need to retreat into the abstract in order to find interesting physical phenomena. Furthermore, his ability to distill the essential information from observations has been the integral force in directing this research to the place where it now stands. The conversations I had with him where always inspiring and very insightful.

I would also like to thank Bruno Eckhardt of the Philipps-Universität Marburg for his support that made it possible for me to pursue research, in the form of an external Masters Thesis, in Göttingen.

Many thanks also to Tobias Lapp for his rigorous experimental work on breath figures. Furthermore I wish to express my gratitude for the conversations we have had about breath figures, his insights into the experiments were crucial for guiding the theoretical investigation. These conversations were of great help. In this context, I also would like to thank Andrew Scullion for his work on the image analysis code, leading to an improvement in the quality of data obtained from the experiment.

I would like to thank Daniel Herde, Frank Male, Mitja Kleider, Jan-Hendrik Trösemeier and Heiko Engelke for their work in proof reading this thesis and their company during the many late nights involved in writing.

Finally I wish to thank my parents, Manfred and Regina Blaschke, for their wisdom, emotional support and patience during a time when I had to invest my full attention and energy towards writing.

7.1 External Sources

The colorful picture of droplets on the cover of this booklet was produced by Andrew Scullion and is derived from an image of the breath figures seen in the experiment of Lapp.

Fig. 1.1(a) and Fig. 1.1(b) were taken by Tobias Lapp whilst cooking. Fig. 1.1(c) was taken from the website: `www.kewlwallpapers.com`. Fig. 1.2 was taken from [7]. Fig. 1.4(b) was taken from their follow-up paper [8]. Fig. 1.3 was supplied by Tobias Lapp, as was Fig. 1.5.

Fig. 1.4(a) is derived from the data of Lapp's experiments. Fig. 2.11 where also taken from the experiments of Lapp.

Fig. 3.2 and Fig. 3.3 where taken from [9].

Appendix A

Calculating circumscribing circles for Triangulations

The fundamental property that every triangulation relies upon is that for any 3 points, one may circumscribe the triangle spanned by the points by 1 unique circle. Due to this fact, for any collection of N points, one may find a unique set of triangles[1] with the property that all points either lie on or outside any circumcircle. This has the wonderful property that we may reduce any triangulation to a set of centers (of circumcircles) for which there are 3 points which are equidistant to these centers, and all others are further away.

A.1 Pedoe's Method for Finding Circumcircles

Pedoe [11] (pp. xii-xiii) gives a simple geometric construction for the circumcircle of the triangle whose vertexes is given by Cartesian coordinates: $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$, then the circumcircle is the set of all Cartesian points $\{(x, y)\}$ satisfying[2]:

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 \tag{1.1.1}$$

this may be expanded to give:

$$a\left(x + \frac{b_x}{2a}\right)^2 + a\left(y + \frac{b_y}{2a}\right)^2 - \frac{b_x^2}{4a} - \frac{b_y^2}{4a} + c = 0 \tag{1.1.2}$$

[1] up to degeneracy incurred due to equidistant points

[2] and the three points lie on this circle since the row vectors are trivially linearly dependent when (x, y) is an element of this set

with:

$$a = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

and:

$$b_x = -\begin{vmatrix} x_1^2 + y_1^2 & y_1 & 1 \\ x_2^2 + y_2^2 & y_2 & 1 \\ x_3^2 + y_3^2 & y_3 & 1 \end{vmatrix}$$

and:

$$b_y = \begin{vmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ x_2^2 + y_2^2 & x_2 & 1 \\ x_3^2 + y_3^2 & x_3 & 1 \end{vmatrix}$$

and:

$$c = -\begin{vmatrix} x_1^2 + y_1^2 & x_1 & y_1 \\ x_2^2 + y_2^2 & x_2 & y_2 \\ x_3^2 + y_3^2 & x_3 & y_3 \end{vmatrix}$$

Thus Eq. (1.1.2) describes a circle with center at $\left(x_0 = -\frac{b_x}{2a}, y_0 = -\frac{b_y}{2a}\right)$ and radius: $r = \frac{\sqrt{b_x^2+b_y^2-4ac}}{2|a|}$.

A.2 Triangulations

Any set of points, may be partitioned into subsets each of 3 points and for which there exists a “center point” such that said 3 points are equidistant and closest to their circumcenter. This is the defining property of a triangulation. This kind of triangulations is called a Delaunay triangulation.

Voronoi [16] used this property to ascribe a set of edges to each point for such a collection, delimit the region closest to the point in question. The center points in the Delaunay triangulation form the vertexes of the Voronoi polygon, with the Voronoi polygon edges being orthogonal to the Delaunay triangulation edges. Thus the Delaunay and Voronoi lattices may be described as being conjugate lattices.

In practice it is very easy to construct the Delaunay lattice, and use this to determine the Voronoi lattice.

Both delaunay and voronoi lattices are most easily described in Fig. 3.7.

Appendix B

Fitting Power-Law Data

Clauset *et al* [6] make an important point regarding the fitting of power-law distributed data. After collecting power-law distributed data, one may not simply take the logarithm of the x and y data-points and apply a simple least squares fitting algorithm to the “linearized” data. On the surface, this results in incorrectly fitted parameters and estimated errors being too small (in some cases being so severely underestimated that the true value of the fitted exponent fall outside the one-σ confidence interval!).

The error in the least-squares fitted exponent results from the fact that least squares fitting algorithms are not constrained by having to produce a result which yields a normalized distribution: in other words, the fitted cumulative density function does not have to reach 1! This allows the least-squares fitting algorithm to make a trade-off between the fitted exponent and lack of a normalized probability density function, producing a “better” fitted exponent (i.e., one which produces a lower error) at the cost of loosing normalization.

This ill-advised trade-off is exasperated by an incorrect estimation of the error in each data-point, as least-squares fitting methods assume independent Gaussian noise.

The log of Gaussian noise, however, is no longer Gaussian! Since the error estimates in standard regression formula rely on the assumption that the data points have independent and Gaussian noise, the estimates for error in the scaling exponents will also be wrong.

This is exasperated furthermore in the case of fitting the cumulative density function using least squares; while fitting cumulative density functions sometimes return better estimates for the exponent, the estimates for the error are in fact worse! This is because the noise in adjacent bins in the cumulative density function are no longer independent, nor Gaussian. Yet the noise is reduced!

Also, since the the noise of each data point is no longer Gaussian, the error in the least-squares fitted exponent is usually underestimated.

Clauset *et al* therefore recommend using the method of greatest likelyhood to fit the scaling exponent. This method is based on maximizing the probability that a "guess" value for the exponent is in fact correct while constraining the function to a normalized form. This method is reportedly extremely successful [6] in determining the exponent of power-law distributed data yielding very accurate results as the normalization of power-law distributed data encodes the scaling exponent also. Ergo, fixing the normalization yields additional information about the scaling exponent.

Sadly though, this method is inappropriate for analyzing the size distribution of our simulated data. When applying Clauset's method, we recover a scaling exponent that can be visually verified as being clearly incorrect. The reason for this is that the power-law distribution in these simulations has an *upper* cutoff as well as a lower cutoff, while Clauset's method only allows for lower cutoffs. As the position of the respective cutoffs does influence the normalization, Clauset's method yields an incorrect result. Their method assumes the wrong normalization.

Hence a more brute-force method is applied in this research. Having established that least-squares is indeed an inappropriate method to use when dealing the linearized non-linear results if this linearization does not preserve the Gaussian-ness of the noise, and that the simple maximum likelyhood estimator method described by Clauset cannot be applied to our data either, we resorted to the Levenberg-Marquardt method. This is a canonical method for performing non-linear fitting without having to first "linearize" the data [12]. The Levenberg-Marquardt method is a iterative method which minimizes the χ^2 value given a function to fit. In this sense it too is a maximum likely-hood method, except that it can fit a wide variety of theoretical functions.

The LM method does require an initial estimate for the fitted parameters. This can usually be made via visual inspection of the plotted data. After estimating the initial scaling exponent, we varied our initial estimate and observed any change in the fitted parameter. This was to ensure that the results produced by the fitting procedure that are stable against error in the initial estimates. Naturally only the results of robust fits are presented here.

Errors in Fitting

When fitting power laws to our data, we assumed an error of approximately $\sqrt{n(s)}$ for each counted number of droplets n. In this case the error in the fitted exponent typically was 0.01, and always below 0.02.

In order to verify robustness, the initial parameters of the Levenberg-Marquardt method were varied. The variation in the fitted exponents also never varied by more than 0.01

upon varying the initial parameters.

Bibliography

[1] J. Aitken. Breath figures. *Nature*, 90, 1913.

[2] D. Beysens. The formation of dew. *Atmospheric Research*, 39(1-3):215 – 237, 1995.

[3] D. Beysens and C. M. Knobler. Growth of breath figures. *Phys. Rev. Lett.*, 57(12):1433–1436, Sep 1986.

[4] B J Briscoe and K P Galvin. Breath figures. *Journal of Physics D: Applied Physics*, 23(9):1265, 1990.

[5] B. J. Briscoe and K. P. Galvin. Growth with coalescence during condensation. *Phys. Rev. A*, 43(4):1906–1917, Feb 1991.

[6] A. Clauset, C. Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *ArXiv e-prints*, June 2007.

[7] Fereydoon Family and Paul Meakin. Scaling of the droplet-size distribution in vapor-deposited thin films. *Phys. Rev. Lett.*, 61(4):428–431, Jul 1988.

[8] Fereydoon Family and Paul Meakin. Kinetics of droplet growth processes: Simulations, theory, and experiments. *Phys. Rev. A*, 40(7):3836–3854, Oct 1989.

[9] Mir Abolfazl Mostafavi, Christopher Gold, and Maciej Dakowicz. Delete and insert operations in voronoi/delaunay methods and applications. *Comput. Geosci.*, 29(4):523–530, 2003.

[10] A.D. Myshkis, V.G. Babskii, N.D. Kopachevskii, L.A. Slobozhanin, and A.D. Tyuptsov. *Low-Gravity Fluid Mechanics.* Springer-Verlag, Berlin, 1987.

[11] D. Pedoe. *Circles: A Mathematical View.* Math. Assoc. Amer. Inc., Washington, DC, 1995.

[12] Willian H. Press, Saul A. Teukolsky, William T. Vetterlin, and Brian P. Flannery. *Numerical Recipes in C++.* Cambridge University Press, Cambridge, 2 edition, 2003.

[13] Rayleigh. Breath figures. *Nature*, 86, 1911.

[14] Rafael Tadmor. Line energy, line tension and drop size. *Surface Science*, 602(14):L108 – L111, 2008.

[15] I. Vardi. *Computational Recreations in Mathematica*. Addison-Wesley, Redwood City, CA, 1991.

[16] G. Voronoi. Nouvelles applications des paramètres continus à la thèorie des formes quadratiques, deuxième memoire, recherche sur les parallelloèdres primitifs. *Journal für die Reine und Angewandte Mathematik*, 134:198–287, 1908.

[17] Henry C. Wente. The stability of the axially symmetric pendent drop. *Pacific J. Math.*, 88(2):421–470, 1980.

ERKLÄRUNG

Hiermit erkläre ich, dass ich meine Masterarbeit mit dem Thema:

Formation and Evolution of
Breath Figures

selbständig verfasst sowie alle wesentlichen Quellen und Hilfsmittel angegeben habe.

Name, Vorname: Blaschke, Johannes

Marburg, den 3. Mai 2010

Einverständniserklärung

Ich erkläre mich damit einverstanden, dass die vorliegende Arbeit

Formation and Evolution of Breath Figures

in Bibliotheken allgemein zugänglich gemacht wird. Dazu gehört, dass sie

- von der Bibliothek der Einrichtung, in der ich meine Arbeit angefertigt habe, zur Benutzung in ihren Räumen bereitgehalten wird,
- in konventionellen und maschinenlesbaren Katalogen, Verzeichnissen und Datenbanken verzeichnet wird,
- der UB für die lokale Benutzung und für Fernleihezwecke zur Verfügung steht,
- im Rahmen der urheberrechtlichen Bestimmungen für Kopierzwecke genutzt werden kann.

Marburg, den 3. Mai 2010

Unterschrift des Autors

Unterschrift des betreuenden Hochschullehrers