Goutte_pendante

An ImageJ plug-in for surface tension measurement through the pendant drop method

Adrian Daerr, Université Paris-Diderot

August 2010, last updated September 16, 2011

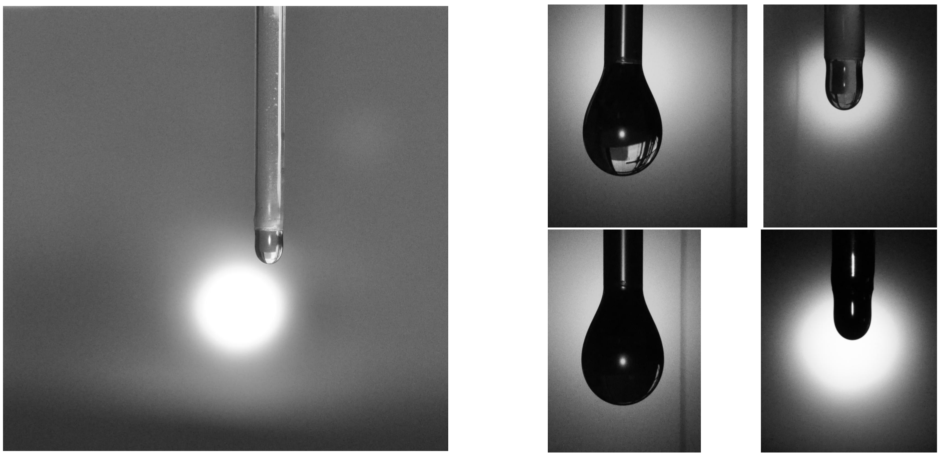

Figure 1: **(left)** Experimental set-up for pendant drop surface tension measurement: a drop is hanging from a capillary tube (diameter here: 1.68 mm). A light source far behind the drop provides a bright background against which the drop is photographed. **(right, top row)** Drops of water and perfluorated oil close to the maximum volume before drop detachment. It is easy to see that the difference in surface tension has a tremendous effect on the drop shape. The pendant drop method consists in exploiting this strong dependency in order to determine an unknown surface tension. **(right, bottom row)** After optimising experimental conditions to get rid of spurious reflections and to enhance contrast.

Abstract

The pendant drop method for surface tension measurement consists in analysing the shape of a drop hanging from a capillary tube (Fig. 1). The `Goutte_pendante` plug-in for ImageJ [1] provides a tool to match a theoretical profile to the contour of a pendant drop, either interactively or automatically. The surface tension can then be easily calculated from the best matching parameters. Section 1 describes the plug-in functionality, section 2 lists the properties that the input image must have for the plug-in to give meaningful results. Sections 3 and 4 explain the installation and the usage of the plug-in. Finally sections 5 and 6 describe the underlying theoretical framework and the plugin's inner workings, respectively.

1 Description

The pendant drop method is commonly used to measure surface tensions of liquids. It consists in analysing the shape of a drop hanging typically from a capillary tube and about to detach (sometimes the inverse situation of a bubble forming at the bottom of a liquid is preferred). The shape is very sensitive to the unknown interfacial tension. The drop profile is described by only one non-dimensional parameter (tip radius over capillary length), although in practice five dimensional parameters can be adjusted within this plug-in: tip position and curvature, tilt of symetry axis and capillary length. The surface tension is calculated from the latter if the density difference is given.

This plug-in allows interactive adjustment of a profile to an image of a pendant drop. An estimate of the quality of the fit is logged to ImageJ's log window. The plug-in can also be asked to improve the fit by varying one or several of the parameters automatically (currently by successive minimisation along chosen directions according to Powell [2, 3], see section 6).

This plug-in can also be used to estimate the surface tension from the shape of a sessile drop.

2 Prerequisites

The plug-in is run on a high contrast image of a pendant drop (Fig. 2-left). Pixel values are expected to be close to zero within the drop, and close to saturation (i.e. 255 for 8-bit images) outside. This is typically obtained by taking the picture of the drop in front of a light background far away from the drop (Fig. 1). Bright spots well within the drop are not a problem, but contrast should be good in the vicinity of the contour. Avoid however thresholded (binary) black and white images, which generally cause the minimisation algorithm to perform poorly. A little bit of blurring on such images will improve the fitting performance.

Draw a rectangular Region Of Interest (ROI) around the pendant drop before calling the plug-in (Fig. 2-middle). The ROI should not include the inlet tube, but only the free

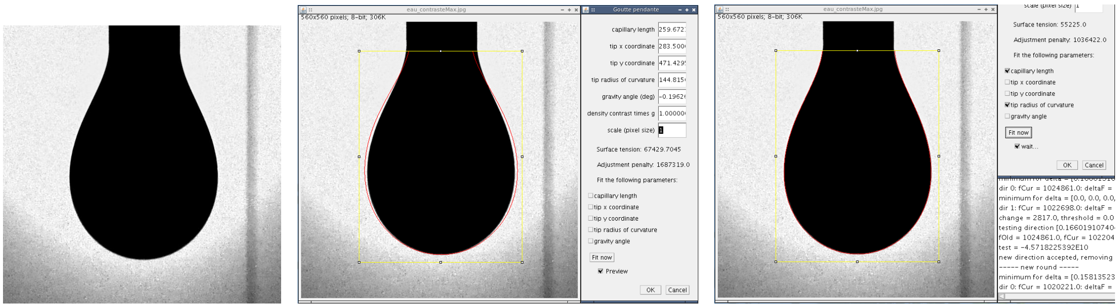

Figure 2: Plug-in usage. **(left)** starting image, high-pass filtered and contrasted just short of saturating black and white levels. **(middle)** image with rectangular ROI selection and initial guess next to plug-in dialog. **(right)** automatic ajustment under way. The progress and decrease of the penalty value is documented in the Log-window in the lower right.

surface of the drop. A few ten or so pixels margin around the drop should be sufficient. Including more of the outside region will not do harm, but slow down calculations which take a time proportional to the area of the ROI.

3 Installing the plug-in

As usual with ImageJ plug-ins, just put the jar into ImageJ's `plugins` folder or a subfolder thereof. Restart ImageJ to see the plug-in.

4 Using the plug-in

Fill in any parameter values you know, and check the plausibility of the initial guess for the others. If the image was calibrated for spatial scale, the corresponding scale is proposed by the plug-in (but beware that ImageJ's dialogs round values to the number of digits shown, so double-check in case your pixel size is much smaller than unity) and all length scales are expressed in calibrated units. If the image is not calibrated all lengths are given in pixels, including the resulting capillary length.

Check the preview box so that the calculated profile is shown as an overlay (Fig. 2-middle).

a) modify the parameters so as to improve the visual accordance of theoretical profile and image. The adjustment penalty value shown in the dialog serves as a quantitative measure of the agreement: it should be made as small as possible.

b) check those parameters that the plug-in may adjust and click on `fit` for automatic minimisation (Fig. 2-right). Only the previously checked parameters are varied when trying to find an optimal shape. The overlay is updated accordingly. Note that the minimisation algorithm can get trapped in a local minimum, so try starting from different initial values and check whether the resulting profile looks satisfactory.

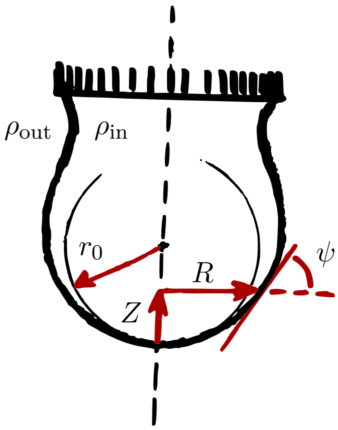

Figure 3: Notations. The drop profile is assumed to be axisymetric, and is described by the two cylindrical coordinates $R(s)$ and $Z(s)$ as a function of the curvilinear parameter s. The local tilt ψ with respect to the axis normal enters the expression for the interface curvature, and is related to R through $R' = \cos\psi$. We have spherical symetry at the very tip, with radius r_0.

To interrupt fitting at any time click once more on the `fit` button.
Exit the plug-in by clicking on either OK or Cancel.

How to get the surface tension If the image was scale-calibrated, and a sensible value was entered for the '`density contrast times g`' parameter (the difference of the densities in/outside the drop multiplied by your planet's gravitational acceleration), then the surface tension is indicated directly in the dialog. For example if the `pixel size` is in millimetres, and the `density contrast times g` is in grams per square second and square millimetre (e.g. around $9.8\,\mathrm{g/mm^2s^2}$ for water in air on earth), then the indicated surface tension is in $\mathrm{g/s^2}$ or milli-Newton per metre. Otherwise the surface tension σ has to be calculated by hand from the indicated capillary length using the formula $\sigma = \Delta\rho g \ell_c^2$.

5 Physics of the pendant drop

This section describes the physics behind the pendant drop method.

Because of the density difference between the inner liquid and outer gas phase, the pressure jump across the interface varies with height, implying that the surface curvature must vary as well. Indeed, if the lowest point of the drop is taken as origin for the height coordinate z, the pressure inside the drop is $p_{\text{in}}(z=0) - \rho_{\text{in}} g z$ at a distance z above it, while it is $p_{\text{out}}(z=0) - \rho_{\text{out}} g z$ outside. g denotes gravitational acceleration and $\rho_{\text{in/out}}$ the densities of the respective phases. Noting $\Delta p_0 = p_{\text{in}}(z=0) - p_{\text{out}}(z=0)$ the pressure drop at the drop tip, and $\Delta\rho = \rho_{\text{in}} - \rho_{\text{out}}$ the density contrast, we have a pressure jump across the interface $\Delta p(z) = \Delta p_0 - \Delta\rho g z$ at height z. Note that the value of Δp_0 is unkown beforehand (it depends on the geometry of our set-up and the drop volume).

This pressure jump is due to surface curvature and surface tension σ. It is related to the mean curvature $\bar{\kappa}$ by Laplace's formula $\Delta p(z) = \sigma\bar{\kappa}(z)$. We parametrise the drop surface in cylindrical coordinates as $(R(s), Z(s))$, where s is the curvilinear distance to the tip. Expressing the mean curvature as a function of R, Z and the angle ψ between the tangent plane to the interface and the horizontal: $\mathrm{d}R/\mathrm{d}s = \cos\psi$, we obtain the following expression for the pressure equilibrium:

$$-\frac{1}{\ell_c^2}\sin\psi = \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\mathrm{d}\psi}{\mathrm{d}s} + \frac{\sin\psi}{R}\right).$$

The boundary conditions at the drop tip are $\psi(0) = 0$ and $\mathrm{d}\psi/\mathrm{d}s = p(0)/2\sigma = 1/r_0 =$ const., where r_0 is the radius of curvature of the drop tip which has locally spherical symetry. Scaling all lengths by the characteristic length $\ell_c = \sqrt{\sigma/\Delta\rho g}$ (called *capillary length*) yields one universal drop shape equation

$$-\sin\psi = \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\mathrm{d}\psi}{\mathrm{d}s} + \frac{\sin\psi}{R}\right) \qquad \text{with } \psi(0) = 0 \text{ and } \psi'(0) = \ell_c/r_0, \tag{1}$$

Note that the solutions (i.e. all possible drop shapes) are fully determined by the choice of the initial condition for $\psi'(0)$, which is just the ratio of the capillary length and the radius of curvature of the drop tip.

6 What the plug-in does internally

numérical integration
The equation (1) is integrated for a given parameter $\ell_c/r_0 = 1/r_*$ using a fourth order Runge-Kutta scheme. Near the drop tip the integral equation becomes singular as $R \to 0$, but the solution itself is perfectly regular: it is simply a nearly perfectly spherical profile (at least at distances small with respect to capillary length and tip radius). We can thus use an analytical solution to get away from the singularity at $R = 0$. The leading order terms of that solution are

$$\left.\begin{array}{rcl} R &=& r_* \sin\frac{s}{r_*} + \frac{s^5}{40r_*^2} + O(s^6) \\ Z &=& \frac{1}{2r_*}s^2(1 + (\frac{1}{16} - \frac{1}{12r_*^2})s^2) + O(s^6) \\ \psi &=& s/r_*(1 - s^2/8) + O(s^5) \end{array}\right\} \text{for} \quad s \ll 1 \wedge s \ll r_*$$

We use this solution to calculate the first piece of drop contour where $s \ll 1$ and $s \ll r_*$. Then we continue through numerical integration.

estimate the qualitiy of the obtained solution
The resulting drop contour is then rescaled, translated and rotated according to the given parameters (tip position, curvature, ...) and overlayed as theoretical contour $\mathcal{C}$ on the drop image. An adjustment penalty value is finally calculated as the sum over all pixels in the Region Of Interest

$$\mathcal{P} = \sum_{\text{pixel } p \in \text{ROI}} \text{Err}(p)$$

of per-pixel-penalties $\mathrm{Err}(p)$, which are determined from the pixel intensity value $I(p)$ as

$$\mathrm{Err}(p) = \begin{cases} I(p) & \text{if } p \text{ is inside the drop contour } \mathcal{C} \\ I_{\max} - I(p) & \text{if } p \text{ is outside the drop contour } \mathcal{C} \end{cases}$$

The rationale behind this penalty function is to count drop and non-drop pixels as penalty for the calculated contour if they are on the wrong side. The input image is such that pixels belonging to the drop on the image are near black, i.e. have a small, near zero intensity value: $I(p) \approx 0$. If they are within the calculated contour as they should their $\mathrm{Err}(p)$ is small according to the formula above, whereas they contribute a lot to the penalty if they are outside $\mathcal{C}$: $\mathrm{Err}(p) = I_{\max} - I(p) \approx I_{\max}$ in this case. Likewise pixels outside the drop belonging to the bright background on the image, near maximum intensity $I_{\max}$, hardly contribute to the penalty function if they are indeed outside the contour, but add their full intensity value to $\mathcal{P}$ if they are inside $\mathcal{C}$.

optimize by varying parameters and iterating
In case automated fitting of some parameters is requested, the plug-in tries to minimise the penalty function by successively varying each free parameter, searching for each the value that will yield the best-fitting integrated drop contour. After all parameters have been varied, the plug-in checks whether it is not more efficient to minimise along another direction in parameter space, i.e. modify several parameters concurrently, and possibly updates the list of directions. The whole process is repeated until the change in the penalty value after a full round becomes too small. For more details see references [2, 3].

References

[1] W. S. Rasband, *ImageJ*, U. S. National Institutes of Health, Bethesda, Maryland, USA, `http://imagej.nih.gov/ij/`, 1997-2010.

[2] M. J. D. Powell, *Computer Journal* **7** (1965) 155

[3] S. Brandt, *Datenanalyse*, BI-Wiss.-Verlag (Mannheim, Leipzig, Wien, Zürich) 3rd edition, ISBN 3-411-03200-6 (1992)