SoftwareX 14 (2021) 100665

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

High-throughput analysis of contact angle goniometry data using DropPy

Michael Julian Orella, McLain Evan Leonard, Yuriy Román-Leshkov, Fikile Richard Brushett *

Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02141, USA

ARTICLE INFO

Article history:
Received 14 March 2020
Received in revised form 25 January 2021
Accepted 25 January 2021

Keywords:
Contact angle goniometry
High-throughput image analysis
Automatic edge detection

ABSTRACT

At present, surface wettability measurements are an underutilized segment of the characterization toolkit, in part due to the redundancy inherent in manual analysis. Even so, there have been numerous advances in contact angle data collection and analysis methods. The emergence of inexpensive and powerful hardware in increasingly small form-factors and the development of robust and versatile software packages would enable interrogation of wetting phenomena across a range of platforms. Here, we introduce DropPy, an open-source Python implementation of the classic axisymmetric drop shape analysis technique to fit droplet profiles from images while providing an easy interface through which casual users may interpret their findings.

Code metadata

Current code version	*v1.0.0b0*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_122
Code Ocean compute capsule	
Legal Code License	*GPL v3.0*
Code versioning system used	*git*
Software code languages, tools, and services used	*Python 3.6*
Compilation requirements, operating environments & dependencies	*SciPy, NumPy, Scikit-image, ImageIO, ImageIO-ffmpeg, PyQt-5, Numba, joblib*
Support email for questions	brushett@mit.edu

1. Motivation and significance

The wetting of a fluid on a solid surface is of great importance in many different technologies, including the development of self-cleaning and other super-hydrophobic materials [1–5], the miniaturization of many analytical techniques and pieces of equipment [6–10], the improvement of optical devices [11–13], and the design of discrete heterogeneous interfaces in reacting systems [14–16]. For example, our group recently demonstrated the relationship between gas diffusion electrode flooding in electrochemical carbon dioxide reduction and reactor failure [17], which suggests a strong correlation between the ability of the liquid electrolyte phase to wet an electrode material and the long-term performance of the system [18]. To design electrodes more aptly for such systems, engineers could perform simple experiments that involve depositing droplets of typical product compositions on a proposed electrode and subsequently examining the dynamic wetting behavior, which would necessitate high-throughput analysis of individual video frames. Further, the availability of accessible and portable surface wetting techniques could facilitate materials characterization in resource-constrained or mobile environments, including laboratories in developing economies or field work in remote locations. Finally, by demystifying the analysis of interfacial phenomena, quantification of fluid wetting on solid surfaces may be incorporated into the general characterization repertoire of the modern researcher.

Unfortunately, to date, the study of materials wetting has largely been confined to research groups with a primary focus on surface chemistry who possess the expertise and equipment necessary to perform detailed characterizations. Given the abundance of detailed methods and hardware available, this is not due to a lack of accessibility of techniques [19]. The open-source software revolution has given rise to a plethora of easy-to-use implementations of image analysis techniques that can be enlisted

* Corresponding author.
E-mail address: brushett@mit.edu (Fikile Richard Brushett).

https://doi.org/10.1016/j.softx.2021.100665

Table 1
Parameters that can be specified at the command line interface for DropPy operation.

Parameter name	Default	Bounds	Meaning
circleThreshold	5	$(0, \infty)$	How many pixels above the baseline edges must be before being considered in the droplet
linearThreshold	10	$(0, \infty)$	The number of pixels inside the circle which can be considered linear
frequency	1	$(0, \infty)$	How often frames should be analyzed (1/fps)
sigma	1	$(0, \infty)$	Initial value for the Gaussian filter used by the Canny algorithm (may be overwritten if –checkFilter is specified)
startSeconds	5	$(0, \infty)$	Burn in period for the droplet to settle
fitType	linear		One of the three fits described above (linear, circular, or Bashforth–Adams)
tolerance	1×10^{-5}	$(0, \infty)$	Step size used for finite difference approximations
maxIters	10	$(0, \infty)$	How many times a circle can be fit to eliminate glare before returning no fit
verticalTolerance	8	$(0, \infty)$	Pixel error for fit that is an indication a vertical fit should be attempted

to rapidly retrieve data from surface wettability images [20,21]. Indeed, over the past few decades, the decreasing cost of image-capture technologies and the increasing availability of powerful computational hardware and software for subsequent image analysis has lowered the barriers to the measurement of fluid wetting [22–29]. Accordingly, we believe that even the casual researcher should be able to extract valuable information from surface wettability studies as part of a broader materials characterization suite. However, modern implementations of contact angle fitting are either simple to install and use but encourage input that may introduce user biases into the results [26,30], or are reliant on expert-level understanding to obtain well-fit data [29]. As a result, most users tend towards time-consuming manual data analysis, where the consequences of image choice and certain parameter selections on their ultimate results are clearer. Such a time-consuming data analysis pipeline precludes the incorporation of dynamic wettability measurements which can be captured by video and may contain a wealth of relevant surface characterization information.

Here we report the development of Droplet analysis with Python (DropPy) [31,32], an open-sourced software package based on the axisymmetric drop shape analysis (ASDA) method [29], which is robust enough to classify contact angle images of droplet profiles on non-reflective surfaces in sessile measurements with minimal user input to avoid bias, and with an interface that provides insight to non-expert users informing their analysis of accurate surface wettability. We first detail the software architecture and theory within the fitting functionalities that are available in DropPy. We then outline the software usage through the assessment of several images on representative planar materials that were collected within our own laboratory. Through the publication of this software on the Python Package Index (PyPI) [32] and its source on GitHub [31], we aim to provide the first step towards reducing the learning curve required for all users of contact angle data. In this way, we envision that through further developments in this space, researchers will be able to robustly analyze multiple heterogeneous data sets in a high-throughput manner without sacrificing understanding, hopefully spurring greater usage of available surface interaction data.

2. Software description

2.1. Software architecture

DropPy performs high-throughput analysis of sessile droplet images according to the algorithmic progression in Fig. 1. The user will specify several parameter values which are summarized in Table 1, but which can be used for all images being processed. First, depending on the file type, DropPy reads the file using `scikit-image` [33] or `imageio` [34] with the `imageio-ffmpeg` [35] plugin active, each available from PyPI. Next, in the sole instance of user interaction with the program, the user can crop the image that has been read into the program to a desired height and width, such that the droplet being analyzed is centered within the resultant image, and can select values that appropriately identify the edges of the goniometer stage and droplet using the Canny edge detection algorithm [20] (**Appendix A**), also available from `scikit-image` [33]. The user can choose to bypass both interactions by specifying the appropriate flags, which will crop the image as described in **Appendix B** and use the specified Gaussian filter value to remove noise from the image before edge detection via the Canny algorithm [20]. For the case of a video file, the subsequent steps will be repeated for each frame to be analyzed within the video, while for a single image file they occur only once. First, the image file read into the program will be converted to grayscale for detection of all edge points, which are separated into points on the baseline and points in the droplet, as shown in Fig. 2. This image is assumed to contain points on the droplet at the user-specified height δ_{circle} above this baseline. Choosing an accurate δ_{circle} will depend on the roughness of the surface and thickness of the baseline points. A rougher surface would warrant a greater δ_{circle} but will cause a corresponding loss of information from the local region of the droplet interface with the solid. As described in **Appendix C**, the baseline is identified using a linear Hough Transform, while the points on the droplet are fit using one of the three approaches which the user must select. These approaches are outlined in Sections 2.1.1–2.1.3 to describe the costs, benefits, and potential use cases of each, so that the user can make an informed decision in selecting the fitting type.

2.1.1. Linear fitting

The coarsest approach for identifying the contact angle from the droplet profile is fitting the portion of the droplet closest to the baseline to a contact line and measuring the angle between the tangent and baselines, as suggested by the definition of the contact angle [36]. Owing to the simplicity of this model, it is extremely sensitive to changes in the points selected for the fitting and must be carefully examined to ensure agreement between tangent line and droplet edge.

To extract the points that appear within a linear profile, the points between δ_{circle} and $\delta_{circle} + \delta_{linear}$ above the baseline are collected. From this subset of linear points on the droplet ($l \subset d$), a standard linear least squares solution can be computed, according to Eq. (1), which fits the vector of measured x-pixel data, $\hat{\mathbf{x}}$, and y-pixel data, $\hat{\mathbf{y}}$, to a line of the form $y = mx + b$, where m is the slope of the line and b is the intercept, using the `numpy.linalg.llsq` implementation [37].

$$\begin{bmatrix} m \\ b \end{bmatrix} = \left(\begin{bmatrix} \hat{\mathbf{x}} & \mathbf{1} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \hat{\mathbf{x}} & \mathbf{1} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{x}} & \mathbf{1} \end{bmatrix}^{\mathrm{T}} \hat{\mathbf{y}} \tag{1}$$

Unfortunately, this fitting approach will fail as the linear points approach verticality (*i.e.* $\theta \approx 90°$), namely when one of the singular values of the matrix $\mathbf{A} = [\hat{\mathbf{x}}\ \mathbf{1}]$ approaches 0. In this case, the functional form of the line $y = mx + b$ becomes inappropriate and instead the points should be fit to a line of the form $x = m'y + b'$, where the parameters can be found in an identical

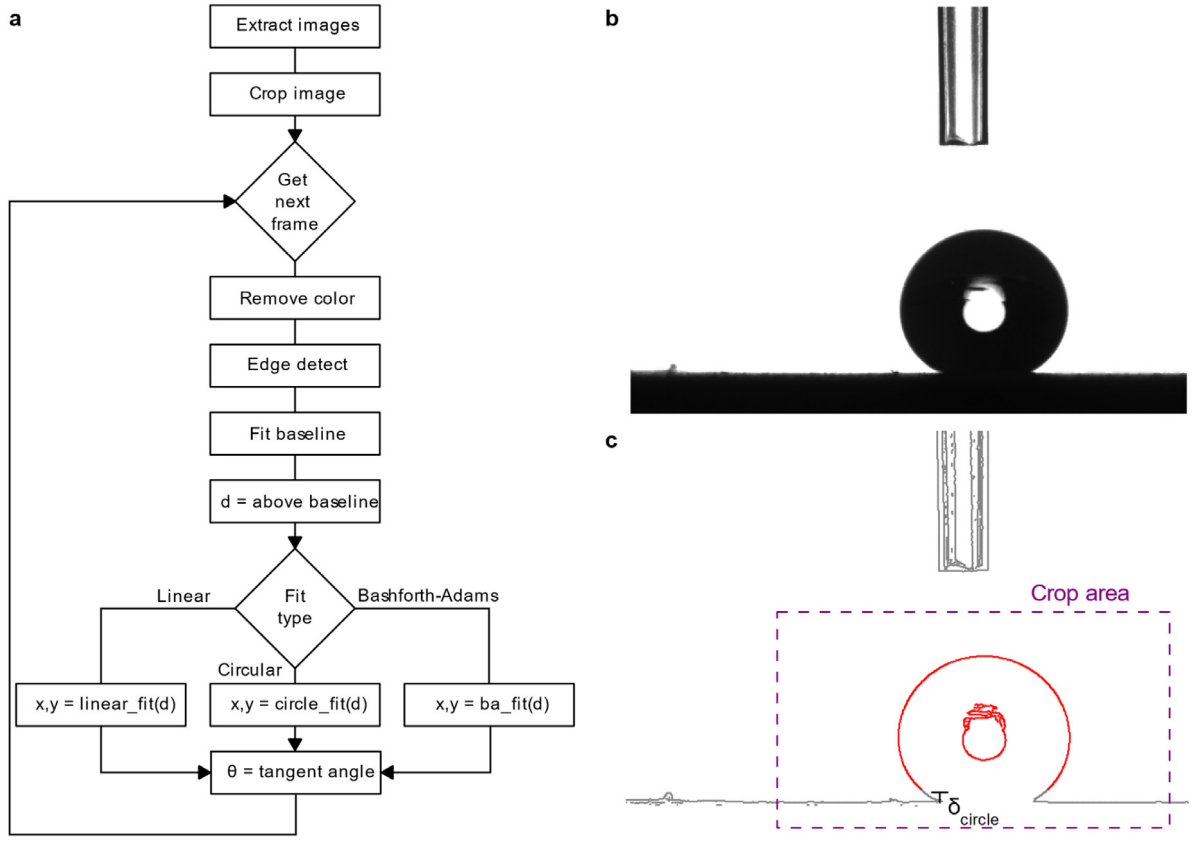

Fig. 1. (a) Logic flow diagram used to automate the analysis of sessile drop goniometry images using either a linear or spherical cap approximation, or a full fitting of the Bashforth–Adams model for droplet profiles. (b) Original example image of a droplet on a surface with a contact angle $\theta > 90°$. (c) Result of the Canny edge detection algorithm processing of the image file with the standard deviation of the Gaussian blur, $\sigma = 0.3$, showing the most important user-specified threshold and crop area calculated.

manner to Eq. (1), with $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ swapped. Once the parameters for these linear fits have been determined, they can be converted to vectors $\mathbf{v} = [1,\ m]^{\mathrm{T}}$ or $[m',\ 1]^{\mathrm{T}}$ if the vertical fit was used and $\mathbf{v}_{\mathrm{b}} = [1,\ m_{\mathrm{base}}]^{\mathrm{T}}$ for the tangent line and baseline. The slope of the baseline, m_{base} is directly calculated from the Hough Transform described in Appendix C. The contact angle is computed from the dot product of these two vectors, as shown by Eq. (2).

$$\theta = \arccos\left(\frac{\mathbf{v}^{\mathrm{T}}\mathbf{v}_b}{\|\mathbf{v}\|\,\|\mathbf{v}_b\|}\right) \tag{2}$$

Fig. 2 summarizes this procedure graphically, showing first the separation of the droplet points into linear and non-linear points, and then the eventual fitting using the linear least squares method.

2.1.2. Circle fitting

The second approach is a model that takes advantage of the full droplet profile by approximating the shape as a spherical cap [38], calculates the points of intersection between the sphere and baseline analytically, and computes the contact angles from the points of intersection. The model is derived from the Young–Laplace equation which assumes that the gravitational forces do not meaningfully distort the shape of the fluid droplets [39]. As such, this fit faces challenges when the droplet has a contact angle above ca. 150° (sometimes classified as "superhydrophobic" [2]) or when gravitational stresses are large compared to the surface tension on the interface, but performs faster than the complete Bashforth–Adams model as there is no initial value problem to solve (*vide infra*). To implement this approach, we attempt to fit the entire droplet edge profile to a circle of the form $(x - z_0)^2 +$

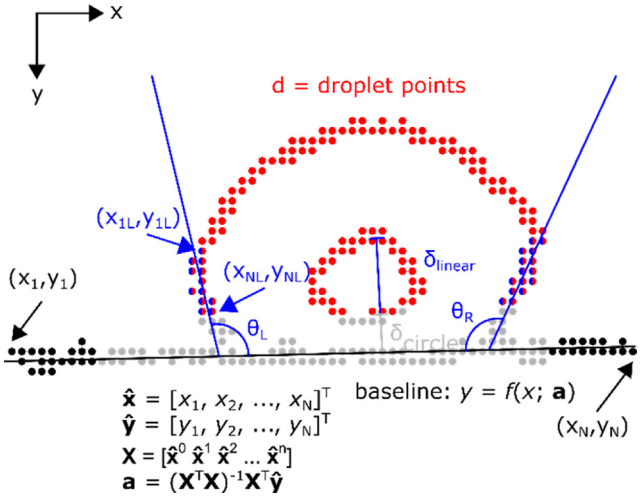

Fig. 2. Graphical description of the approach used to fit the baseline and droplet edge to a linear profile. In this diagram, $\delta_{\mathrm{baseline}}$ refers to the width from the edge of the cropped image which only contains points on the baseline, highlighted in black, while any points within δ_{circle} pixels of the baseline are assumed to be a part of the baseline, as shown in gray. The red points are those greater than δ_{circle} pixels above the baseline, while the blue and red points are those assumed to be in the linear regime between δ_{circle} and $\delta_{\mathrm{circle}} + \delta_{\mathrm{linear}}$ to which the linear fit is applied. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$(y - z_1)^2 = r^2$, where $\mathbf{z} = [z_0,\ z_1]$ is the vector to the center point of the droplet, and r is the radius of the droplet. To fit

a circle to the set of points that define the edge of the liquid droplet, we sought to minimize the total distance between the experimentally observed points and the circle [40], as shown by Eq. (3), where $\mathbf{x}_i$ is the vector of the ith (x, y) coordinate of a point on the droplet edge.

$$\mathbf{z}^*, r^* = \arg\min_{\mathbf{z},r} \sum_{i=1}^{N} (\|\mathbf{z} - \mathbf{x}_i\| - r)^2 \tag{3}$$

This optimization is accomplished using `scipy.optimize.minimize` [41,42] with the default Broyden–Fletcher–Goldfarb–Shanno method [43], an initial estimate of the droplet radius as half the image width, and the estimate for the center point as the mean of the x, y data representing the droplet edges. While this optimization can robustly identify the best-fit circle, some droplets will have interior glare that must be removed, which we accomplish through an iterative fitting approach. If the optimal parameters obtained from fitting all of the points in d (Fig. 3a) result in sum-squared error that was greater than the number of points, N, detected above the baseline, the optimization would execute again, keeping only the points outside the first best-fit circle (Fig. 3b).

Once the parameters describing the circle that best fit the droplet have been obtained, the points of intersection between the circle and baseline can be used to calculate the contact angles, but the baseline may have a non-zero slope. Instead, we utilized straightforward coordinate transformations to shift the baseline to a horizontal line before proceeding with computing the points of intersection. In case of a horizontal baseline, it can simply be represented by $y' = B$, where B is the constant height of the line of intersection in the transformed coordinate system, and the x-coordinate of intersection can be calculated as $x' = \pm (r^2 - B^2)^{1/2}$. In this case, the constant B can be determined from the coordinate transformations necessary to achieve a horizontal baseline. As such, our goal is to identify a transformation that maps the original coordinate system (x, y) to a new coordinate system (x', y') where the baseline has zero slope, as shown in Fig. 4. To accomplish this, we first translate the coordinates to be located at the circle center, so that $\tilde{x} = x - z_0$ and $\tilde{y} = y - z_1$. In this translated coordinated system, the circle is simplified to the locus of points that satisfies $\tilde{x}^2 + \tilde{y}^2 = r^2$ and the baseline now satisfies $\tilde{y} = m\tilde{x} + (mz_0 + b - z_1)$, recognizing that this translation has not modified the slope of the line. To remove all dependence of the baseline on the x-coordinate, we rotate the intermediate coordinate system $(\tilde{x}, \tilde{y})$ about the origin by an angle θ defined such that $\tan(\theta) = m$. We can define the new coordinate system by the rotation given in Eq. (4).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} \tag{4}$$

From this rotation matrix, the equation for the baseline can be solved in terms of (x', y'), to obtain the result shown in Eq. (5), which, as expected, has no dependence on x', the final x-coordinate.

$$y' = \frac{mz_0 + b - z_1}{\sqrt{1 + m^2}} \tag{5}$$

Once the point of intersection $I((r^2 - B^2)^{1/2}, B)$ has been calculated through the coordinate transformation procedure specified, the vector representing the baseline is identically $[-1, 0]$, while the vector representing the tangent line to the droplet at this point can be computed by $[1, -(r^2 - B^2)^{1/2}B^{-1}]$, if $B \neq 0$, and the angle between them can be calculated according to Eq. (2).

2.1.3. Bashforth–Adams fitting

Finally, the most rigorous model fits the entire droplet profile to the Bashforth–Adams equations [44], which describe the shape of a droplet on a surface while accounting for the effects of gravity, but incurs the greatest computational cost as it requires the numerical solution of ordinary differential equations at each iteration of the optimization. Unless the user requires high-volume analysis, we recommend that they always utilize the Bashforth–Adams approach as it will yield the most quantitatively accurate results. The Bond number (Bo), a dimensionless ratio of gravitational and surface tension stresses, is employed to determine the importance of gravity on the droplet shape. For a droplet with known physical properties and size, the Bo can be defined as the square of the ratio of the radius of curvature at the apex to the capillary length [38]. If Bo $= (\rho g b^2)/\gamma \gtrsim 1$, the Young–Laplace equation should be modified to include pressure variation as a function of height, as classically described by the equation derived by Bashforth and Adams (Eq. (6)) [44]. In this formulation, we describe the position of the loci of points $A(x, z)$ on the droplet profile, where z' is the derivative of z with respect to x at the point A, z'' is the second derivative with respect to x at that point, and b is the radius of curvature at the origin. The interfacial and fluid properties including surface tension between the two fluid phases γ and droplet density ρ dictate the overall shape of the droplet on the planar surface.

$$\gamma \left\{ \frac{z''}{\left(1 + z'^2\right)^{3/2}} + \frac{z'}{x\left(1 + z'^2\right)^{1/2}} \right\} = \frac{2\gamma}{b} + \rho g z \tag{6}$$

While this equation can be solved given the initial conditions $z(0) = 0$, and $\lim_{x\to 0} \frac{1}{x}\frac{dz}{dx} = 1$, it is more commonly parameterized by using the substitution $\tan(\varphi) = z'(x)$ to avoid the inconvenient manipulations necessary as z is not a single-valued function of x [7]. Upon parameterization with the tangent angle φ and introduction of the capillary length $a = (\gamma/(\rho g))^{1/2}$, a set of equations containing only the Cartesian coordinates z and x with the two parameters a and b can be derived, shown in Eq. (7).

$$\frac{dx}{d\phi} = \frac{bx\cos\phi}{a^2 bxz + 2x - b\sin\phi}, \qquad \frac{dz}{d\phi} = \frac{bx\sin\phi}{a^2 bxz + 2x - b\sin\phi} \tag{7}$$

While these differential equations can be solved for the entire curve that defines $z(x)$, there remains a pole at $x = 0$ that prevents integration over the entire domain. As such, we divide the domain in two sections, such that $\varphi \in [-\theta_l, 0)$ or $\varphi \in (0, \theta_r]$, where θ_l and θ_r are the contact angles at the left and right side of the droplet, respectively. The initial conditions used for these domains are $x(0) = -1 \times 10^{-5}$ and $x(0) = 1 \times 10^{-5}$, respectively, while $z(0) = 0$ for both. From these initial conditions, the equations could be integrated towards the baseline using the backward differential formula (BDF) solution method [45] as implemented in `scipy.integrate.solve_ivp` [41]. We measured the error in the solution as the sum of the minimum distance between all pairwise combinations of the predictions generated from the BDF solver, denoted by $[x_i, z_i]^T$, and the points which had been extracted from the droplet profile, denoted by $[x_i, z_i]^T$, using the `scipy.spatial.distances.cdist` [41] function. To compute the best fit contact angles, we minimized this objective function over the two model parameters, as shown below.

$$a^*, b^* = \arg\min_{a,b} \left(\sum_{i=1}^{n} \min_j \left\| \begin{bmatrix} x_i \\ z_i \end{bmatrix} - \begin{bmatrix} x_j \\ z_j \end{bmatrix}' (a, b) \right\|_2^2 \right) \tag{8}$$

While the value of a^*, the best-fit capillary-length for the droplet, can contain significant insight by allowing the measurement of the fluid density and interfacial surface tension, here we primarily focus on the extraction of the contact angle, the parametric

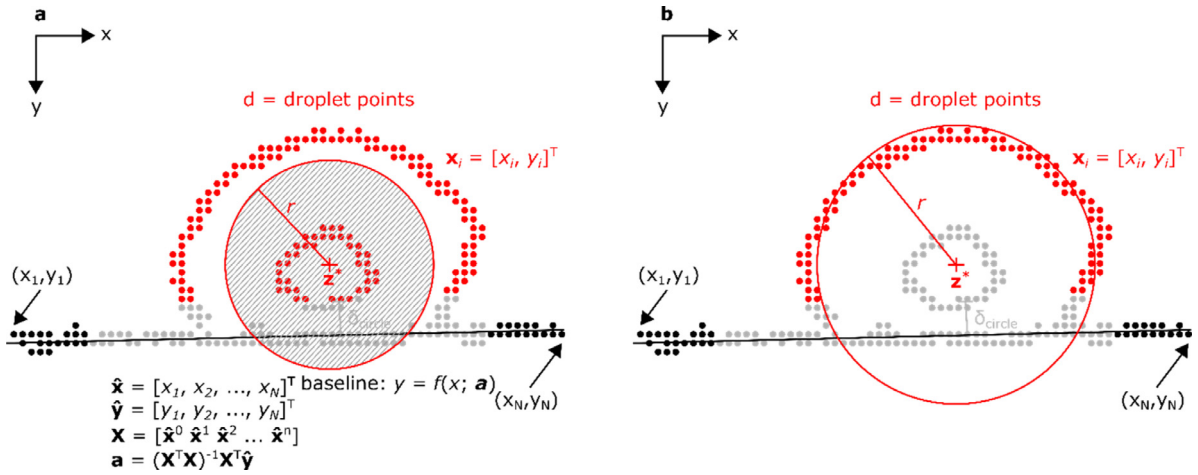

Fig. 3. Procedure of finding the best-fit circle to the droplet edge by eliminating points caused by droplet glare. In (a) all points at least δ_{circle} above the baseline are included in the fit, resulting in a best-fit circle between the inner and outer circles. These points are then excluded from the fit in (b) resulting in a much better fit to the true droplet profile.

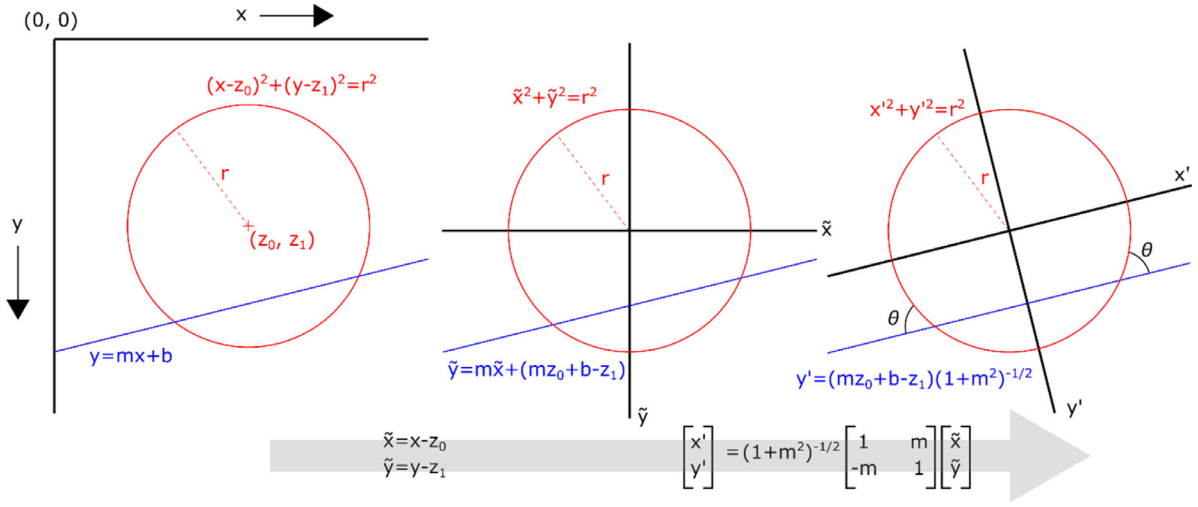

Fig. 4. Coordinate transformation used for the circular fitting approach. The original coordinates show the approximated circle in red and intersecting baseline in blue. These coordinates are then translated and finally rotated to the final coordinate system (x', y') where the baseline is a constant height from the x'-axis. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

angle φ at which the droplet contacts the baseline. The process for extracting the pixels that constitute the edge of the droplet is similar to that used in the case of fitting the spherical cap, and indeed in this case the spherical cap approximation is used to remove any interior glare. Again we used the `scipy.optimize.minimize` [41] function to perform the optimization specified by Eq. (8), but in this case we opted for the Nelder–Mead simplex algorithm to minimize the number of finite derivative estimates that were needed [46,47]. In this solution, the contact angle θ is defined as the parametric angle where the vertical position equals the measured droplet height, or $z(\theta) = H$. As shown by Fig. 5, the droplet profile may fit the circular approximation well in the upper hemisphere, but may deviate significantly as it approaches the baseline, leading to an underprediction of the true contact angle, which can be more accurately estimated from the Bashforth–Adams fit.

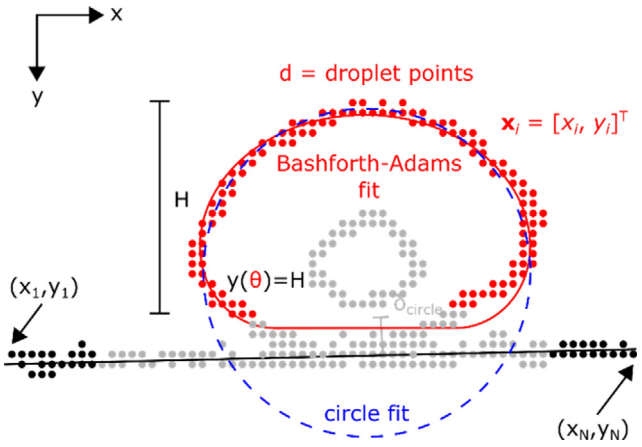

Fig. 5. Comparison between the Bashforth–Adams fitting (red line) and the spherical cap approximation (blue dashed line), showing the improved accuracy of the Bashforth–Adams fit near the baseline as compared to the circular fit.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Software functionalities

DropPy was developed in Python 3.7 enabling it to run on any platform where Python virtual environments exist. The software can be easily installed into a Python environment with Python

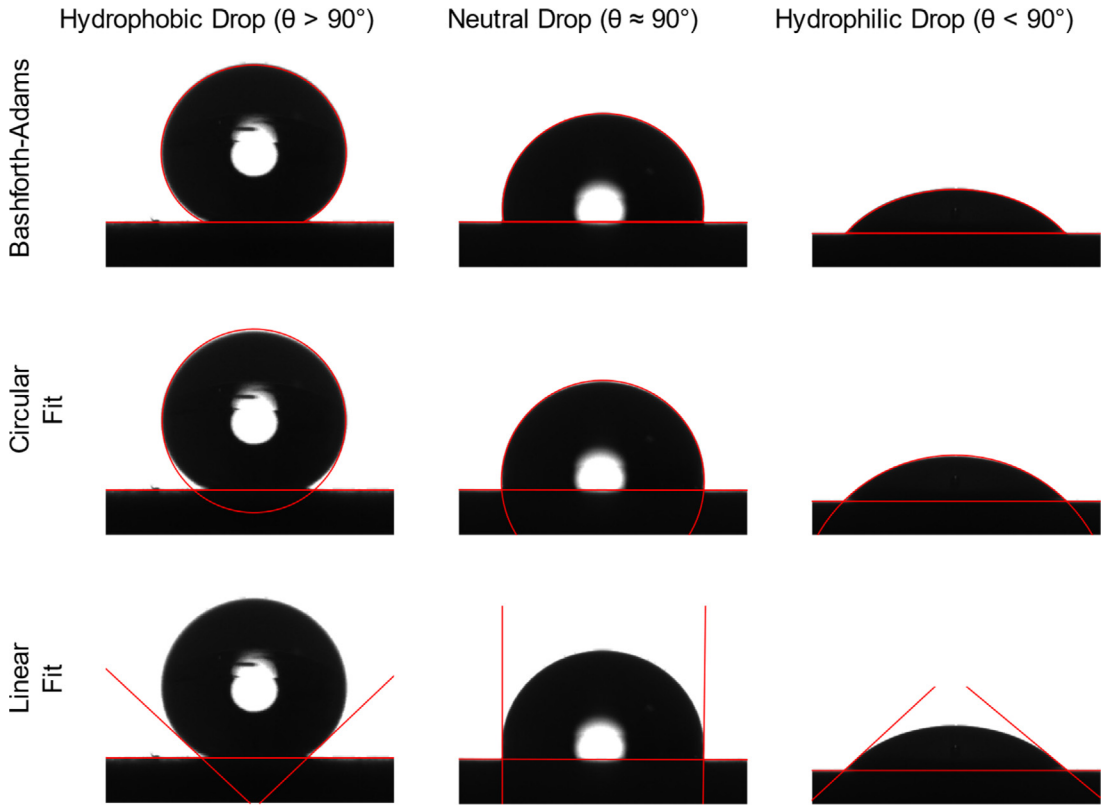

Fig. 6. Example of the droplet fits that are automatically generated using the software package which we developed. These are displayed to the user for visualization of the goodness of fit throughout program operation so that the user can evaluate whether a certain image/video file needs to be re-analyzed. The hydrophobic droplet was composed of 25 wt% formic acid (reagent grade >95%; Sigma-Aldrich) in deionized water, deposited on a Porex PMV15T sheet. The neutral drop was composed of 25 wt% formic acid (reagent grade >95%; Sigma-Aldrich) in deionized water, deposited on a Goodfellow PTFE FP303050 sheet. The hydrophilic drop was 25 wt% 1-propanol (ACS reagent grade >99.5%; Sigma-Aldrich) in deionized water, deposited on a Goodfellow PTFE FP303050 sheet. All of the droplets are 5 μL in volume.

$\geq$ 3.6 using `pip install droppy` as the package is hosted on PyPI [32]. Users can then call the analysis script on individual files or directories that need to be analyzed, referring to one of the three main fitting approaches that should be used. The primary source of user interaction with the script occurs at the terminal interface, where users can specify any of the parameters shown in Table 1.

3. Illustrative examples

Here, we examine the three possible fit types using three distinct regimes of intrinsic surface wettability: "hydrophobic" (θ > 90°), "neutrally wetting" ($\theta \approx$ 90°), and "hydrophilic" (θ < 90°) within the context of a cylindrical capillary. After the user installs the software, the main analysis script can be run by calling the following commands from within the Python environment:

```
$ droppy ./hydrophilic_drop.avi --startSeconds 10 --frequency 1 --autoCrop --fitType {circular|linear|Bashforth--Adams }
```

```
$ droppy ./neutral_drop.avi --startSeconds 10 --frequency 1 --autoCrop --fitType {circular|linear|Bashforth--Adams }
```

```
$ droppy ./hydrophobic_drop.avi --startSeconds 10 --frequency 1 --autoCrop --fitType {circular|linear|Bashforth--Adams }
```

The quality of the fit for all frames analyzed is displayed at the end of analysis procedure as shown in Fig. 6, which can help the user to identify if certain files need reanalysis. As discussed, the linear fit incurs the lowest cost, but appears to coarsely approximate the contact angle for all droplets shown below. Despite the speed, there is significantly greater variance than in the case of the circular fit, as seen in Fig. 7. The droplets which have been fit with the spherical cap approximation appear to have the lowest variance over the entire data set but, as expected, break down for hydrophobic surfaces. In each droplet we examine here, the Bashforth–Adams model predicts the largest contact angle, and qualitatively shows the closest agreement to the droplet shape near the baseline. For this reason, we recommend using the Bashforth–Adams approach to analyze any images unless coarser approximations can be tolerated, in which case the linear fit may be acceptable, or the droplets are known to have Bo << 1, where the spherical cap approximation will work.

From the quantitative results in Fig. 7, we see agreement with the initial classification of the droplets being "hydrophobic", "neutrally-wetting", and "hydrophilic" as expected. For each of these droplets, there is a slight decrease in contact angle over the length of the video, which we hypothesize is fluid evaporation from the droplet. This does not appear to impact the operation of the software. From these data, the underprediction of the hydrophobic contact angle with the linear and spherical cap approximations is immediately apparent, as both predict contact angles nearly 20° below the contact angle measured through the Bashforth–Adams fit. Additionally, we observe a smoothing effect caused by the spherical cap approximation where the measured contact angles within a single drop appear to have less variance as most local information is ignored by fitting to a circular profile. While both the Bashforth–Adams and linear fit take more of the

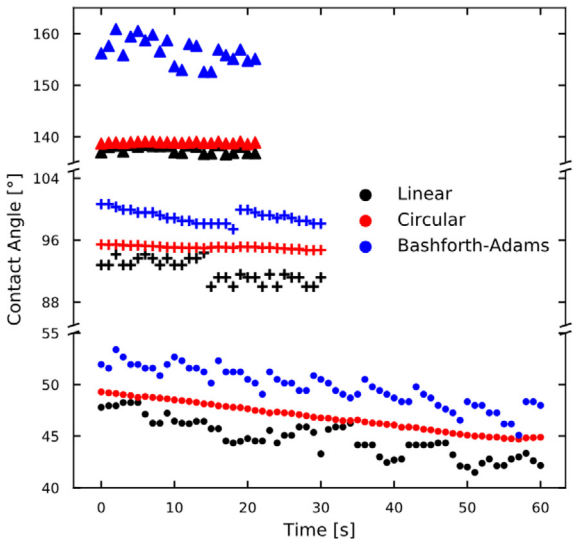

Fig. 7. The results of analyzing the 3 demonstration video files – "hydrophilic" (•), "neutral" (+), and "hydrophobic" (▲) – using the 3 different fitting approaches, as shown by the different colors. These data utilize the same experimental conditions as described in Fig. 6, where the dynamics are obtained from analyzing the entire video rather than a still image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

local profile into account, as evidenced by the larger variance for each of these data sets, the linear fit will inherently miss any of the droplet behavior that occurs between the baseline height and δ_{circle} as potentially being part of the baseline. Although these points are also ignored by the Bashforth–Adams fit, they can be better predicted through the physical nature of the model, which results in a much closer match to the true contact angles.

4. Impacts and conclusions

To provide the most robust software and to maximize utility to the community, DropPy was implemented with three important functionalities. First, DropPy performs an automatic baseline detection to separate the image file into elements that are within the droplet and elements that reside on or near the baseline. Second, is the option for minimal user input which can help eliminate user bias by performing an automatic cropping of the droplet image. Finally, we included three different models that could be used for extracting the contact angle from the droplet profile. By incorporating each of these aspects, we address the targeted goals for DropPy of (1) ensuring multiple types of droplets and surfaces can be well-handled by the software, (2) minimizing user input, and (3) providing users with an understanding of fit quality and how it may be improved. Ultimately, we hope that this software will be useful to a diverse set of researchers expanding the use of contact angle goniometry measurements across multiple fields. To this end, our group has used DropPy to systematically characterize the interactions between droplets of aqueous-organic liquid mixtures and solid surfaces, including graphite, polytetrafluoroethylene (PTFE), and oleophobic PTFE, helping us to better understand wettability constraints for carbon dioxide electrolyzer electrodes [48].

Despite the hardware advances that would support improving the utilization of wetting techniques, challenges with software obscurity and lack of automation hinder the incorporation of surface wettability with a broad suite of routine materials characterization methods. As such, we envision that with the development of DropPy as an open-sourced image analysis toolkit, we can decrease the barrier to entry for the casual user to investigate the effects of surface wetting in their own research areas of interest, without the need for domain specific knowledge. In this way, researchers may leverage the large volume of data afforded through simple contact angle goniometry measurements.

CRediT authorship contribution statement

Michael Julian Orella: Conceptualization, Methodology, Software, Writing - original draft. **McLain Evan Leonard:** Investigation, Validation, Writing - review & editing. **Yuriy Román-Leshkov:** Resources, Supervision, Funding acquisition, Writing - review & editing. **Fikile Richard Brushett:** Resources, Supervision, Funding acquisition, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the National Science Foundation, USA Graduate Research Fellowships Program under Grant No. 1122374 for support of M.J.O. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additionally, this work was supported by research funding from the Massachusetts Institute of Technology, USA. The authors thank Michael Stone and Antoni Forner-Cuenca for insightful discussion and constructive feedback.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2021.100665.

References

[1] Gu Z-Z, Uetsuka H, Takahashi K, Nakajima R, Onishi H, Fujishima A, Sato O. Angew Chem, Int Ed 2003;42:894–7.
[2] Feng L, Li S, Li Y, Li H, Zhang L, Zhai J, Song Y, Liu B, Jiang L, Zhu D. Adv Mater 2002;14:1857–60.
[3] Feng L, Zhang Y, Xi J, Zhu Y, Wang N, Xia F, Jiang L. Langmuir 2008;24:4114–9.
[4] Tuteja A, Choi W, Ma M, Mabry JM, Mazzella SA, Rutledge GC, McKinley GH, Cohen RE. Science 2007;318:1618–22.
[5] Li X-M, Reinhoudt D, Crego-Calama M. Chem Soc Rev 2007;36:1350–68.
[6] Cho Sung Kwon, Moon Hyejin, Kim Chang-Jin. J Microelectromech Syst 2003;12:70–80.
[7] Srinivasan V, Pamula VK, Fair RB. Lab Chip 2004;4:310–5.
[8] Wheeler AR. Science 2008;322:539–40.
[9] Abdelgawad M, Wheeler AR. Adv Mater 2009;21:920–5.
[10] Miller EM, Wheeler AR. Anal Bioanal Chem 2009;393:419–26.
[11] Hayes RA, Feenstra BJ. Nature 2003;425:383–5.
[12] Mugele F, Baret J-C. J Phys: Condens Matter 2005;17:R705–74.
[13] Berge B, Peseux J. Eur Phys J E 2000;3:159–63.
[14] Forner-Cuenca A, Manzi-Orezzoli V, Biesdorf J, Kazzi ME, Streich D, Gubler L, Schmidt TJ, Boillat P. J. Electrochem. Soc. 2016;163:F788–801.
[15] Gostick JT, Ioannidis MA, Fowler MW, Pritzker MD. J Power Sources 2009;194:433–44.
[16] Eikerling M. J Electrochem Soc 2006;153:E58–70.
[17] Leonard ME, Clarke LE, Forner-Cuenca A, Brown SM, Brushett FR. ChemSusChem 2020;13:400–11.
[18] Weng L-C, Bell AT, Weber AZ. Phys Chem Chem Phys 2018;20:16973–84.
[19] Herminghaus S. J Phys: Condens Matter 2005;17:S261–4.
[20] Canny J. IEEE Trans Pattern Anal Mach Intell 1986;PAMI-8:679–98.
[21] Shokhan MH. Int J Adv Eng Technol 2014;7:59–65.
[22] Chandler JH, Fryer JG, Jack A. Photogramm Rec 2005;20:12–26.
[23] Chen H, Muros-Cobos JL, Amirfazli A. Rev Sci Instrum 2018;89:035117.

[24] Chen H, Muros-Cobos JL, Holgado-Terriza JA, Amirfazli A. Colloids Surf A 2017;533:213–7.
[25] Soon CF, Omar WIW, Nayan N, Basri H, Bt. Narawi M, Tee KS. Proc Technol 2013;11:487–94.
[26] Stalder AF, Kulik G, Sage D, Barbieri L, Hoffmann P. Colloids Surf A 2006;286:92–103.
[27] Stalder AF, Melchior T, Müller M, Sage D, Blu T, Unser M. Colloids Surf A 2010;364:72–81.
[28] Hansen FK. J Colloid Interface Sci 1993;160:209–17.
[29] Rotenberg Y, Boruvka L, Neumann AW. J Colloid Interface Sci 1983;93:169–83.
[30] Brugnara M. Contact angle. 2006, https://imagej.nih.gov/ij/plugins/contact-angle.html.
[31] Orella M. Michaelorella/Droppy. 2020, https://github.com/michaelorella/droppy.
[32] Orella MJ. Droppy 1.0.0b0: # Python contact angle image processing analysis. 2020, https://pypi.org/project/droppy/.
[33] van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T. PeerJ 2014;2:e453.
[34] Klein A. Imageio. 2014, http://imageio.github.io/.
[35] Klein A. Imageio/Imageio-Ffmpeg, imageio. 2020, https://github.com/imageio/imageio-ffmpeg.
[36] Marmur A, Della Volpe C, Siboni S, Amirfazli A, Drelich JW. Surf Innov 2017;5:3–8.
[37] Oliphant T. A Guide To NumPy. USA: Trelgol Publishing; 2006.
[38] Deen WM. Analysis of Transport Phenomena. 2012, p. 249.
[39] Young T. Philos Trans R Soc Lond 1805;95:65–87.
[40] Gander W, Golub GH, Strebel R. BIT 1994;34:558–78.
[41] Oliphant TE. Comput Sci Eng 2007;9:10–20.
[42] Millman KJ, Aivazis M. Comput Sci Eng 2011;13:9–12.
[43] Nocedal J, Wright SJ, editors. Numerical Optimization. New York, NY: Springer; 2006, p. 135–63.
[44] Bashforth F, Adams JC. An Attempt To Test the Theories of Capillary Action By Comparing the Theoretical and Measured Forms of Drops of Fluid. with an Explanation of the Method of Integration Employed in Constucting the Tables Which Give the Theoretical Forms of Such Drops. Cambridge [Eng.] University Press; 1883.
[45] Byrne GD, Hindmarsh AC. ACM Trans Math Software 1975;1:71–96.
[46] Nelder JA, Mead R. Comput J 1965;7:308–13.
[47] Wright M. Numerical analysis. In: Proceedings of the 1995 dundee biennial conference in numerical analysis; 1996. p. 191–208.
[48] Leonard ME, Orella MJ, Aiello N, Román-Leshkov Y, Forner-Cuenca A, Brushett FR. J Electrochem Soc 2020;167:124521.