

## Methods

# 1 Overview

Mainly we are concerned about what parameters are important in electrostatic transport of relatively large (e.g. millimetric) droplets in low-gravity, and what the values of the respective dimensionless groups, namely  $\mathbb{I}_m$ ,  $\mathbb{E}_u$  and  $\mathbb{E}_{u+}$  might be.

To find typical values of these parameters we used the following approach:

1. We observed spontaneous droplet jumps on charged dielectric super-hydrophobic surfaces under low-gravity conditions in a 2.1 s drop tower, while varying the independent variables  $V_d$ ,  $\sigma$ . However, this raises an interesting challenge; it is impractical to directly measure all of the key physical quantities that appear in the dimensionless groups at once in a drop tower experiment. In particular, determination of net droplet free electric charge  $q$ , is difficult (as high-input resistance electrometers, being notoriously fickle instruments, are not well suited to sudden 15-g decelerations). Schematic representations of the droplet jump, and of the droplet in the Cassie-Baxter 1-g initial state is shown in Figures 2 and 1.
2. Using high-speed video and image analysis software we captured the trajectories of the droplets.
3. We solved the inverse problem to find the key parameters by maximizing a statistical likelihood function between an observed trajectory and the trajectory predicted by a dynamical model given that certain set parameters. The best fit parameters obtained by direct-search optimization

are those corresponding to the maximum likelihood experimental values. The optimization is constrained by the measurement precision of directly measured independent, and dependent variables.

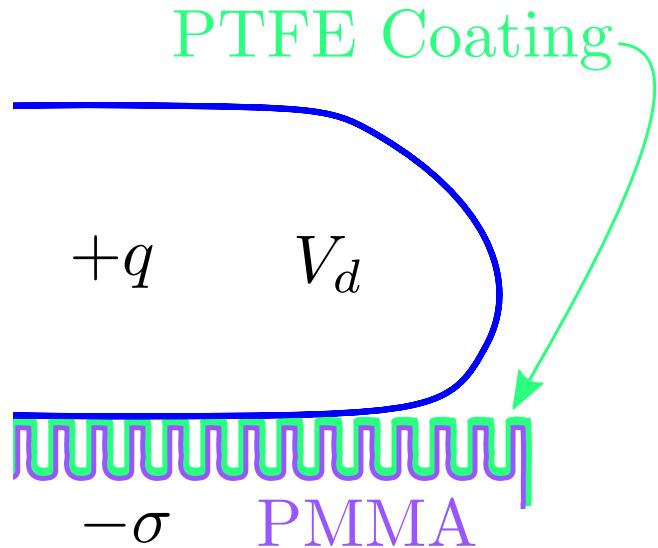


Figure 1: A schematic representation of the droplet initial condition, prior to the drop.

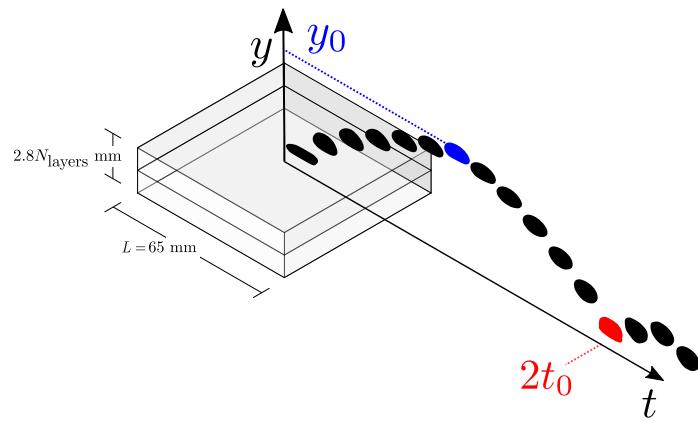


Figure 2: The experimental concept with characteristic time and length scales,  $t_c, y_c$  shown.

## 2 Experimental Methods

Droplets of distilled water, in a range of volumes ( $0 \leq R_d \leq 1\text{mL}$ ), were very carefully deposited on the charged superhydrophobic surface using an grounded glass syringe with  $\pm 1\mu\text{L}$  accuracy, and then dropped in the drop tower. The DDT uses a dual capsule design, inspired by the 2.2 s facility at NASA Glenn Research Center, which decouples drag acceleration felt by the external drag shield from the experiment. The experiment then experiences approximately  $1 \times 10^{-6}\text{g}$  of high quality free-fall for the 2.1s required for the rig/drag shield assembly to plummet to the bottom of drop tower 6 floors below. Red dye was added to improve trajectory digitization. Droplet trajectories were recorded using a *Panasonic HC-WX970 Camera*, at 120 fps and 1/3000s shutter speed. In a few cases where higher frame rates were needed a *Nikon 1 J1 camera* with a 30mm telephoto lens was used, shooting at 400 fps. The experimental test cell was illuminated with a 6000K LED strip, with a thin semi-opaque plastic film covering to make the light diffuse.

Rather than haphazardly contact charge the surfaces with streams of water, as likely happened by accident in the experiments which inspired this inquiry, we strive for a more controlled, uniform surface charge distribution. Following from this desire, superhydrophobic electrets were prepared, with surface potentials 0.7-4.0 kV. According to a review by Sessler [1], an electret is a dielectric material with “quasi-permanent” electric charge in the sense that the characteristic decay period of the electret is much greater than a practical experiment time scale. Electret charge may be ‘true’ charge in the form of surface or space charges, or polarization charges (such as bound charges). If the electret is not screened by a conductor then it produces an external electric field if the polarization and real charges do not uniformly compates each other throughout the volume of the electret. For this reason electrets are though of as electrostatic analouges to permanent magnets (and the name *electret* itself is a portmantau to that effect conjured by Heaviside in 1892 [2]). Typical commerical electrets

are Teflon type (PTFE) polymer films on the order of 10-50  $\mu\text{m}$  thick with the charge being primarily real surface charge. Electrets have a plethora of applications, but most germanely they have been used in Electro-Wetting On Dielectric (EWOD) devices for low-voltage manipulation of small droplets [3]. Real charge electrets can be produced by contact electrification, injection or deposition) of charge carriers by discharge or electron beam, ionizing radiation, or by frictional triboelectrification. Dipolar electrets by contrast are produced by a combination of polarization at elevated temperatures in a strong external electric field, followed by an annealing process. Effective surface charge densities are limited to the material dielectric strengths due to internal dielectric breakdown phenomenon (this typically occurs before external breakdown or Paschen breakdown). We use an isothermal electret formation process using a variation of the widely applied corona-charging technique. The typical corona-charging technique uses strong inhomogenous DC electric fields to produce discharge in air at ambient conditions; the dielectric substrate is atop a grounded electrode, and there is a screening potential electrode intermediately positioned to control the surface potential (the the surface potential of the dielectric will tend to saturate at this grid potential if the material is not space-charge current limited). The corona field is usually produced by pin-shaped electrode. In air the most common charge carriers thus produced are  $\text{CO}_3^-$  ions. This approach is known to generally produce samples with fairly uniform surface charge densities. Some work by van Turnhout in 1975 showed, using Thermally Stimulated Current (TCS) measurements, that in 4.8 mm thick polymethyl methacrylate (PMMA) polarization of the dielectric is non-uniform, due to real space-charge mostly ( $\sim 90\%$ ) residing in a thin (0.1-0.2 mm) layer near the free-surface of the sample [4]. In this work we use a balanced AC corona ion source (Ptec IN5120), to direct a net neutral stream of ions towards the dielectric target, which we polarize by an electrode with a 2 kV+ (absolute reference) DC-DC converter. The ion stream compensates the surface, and space bound charges arising due to the polarization of the dielectric. When the DC-DC converter is switched to

ground, the deposited negative ions remain on the surface.

The electret is lamination of 3-4 0.4 mm corona charged PMMA sheets. The electric field strength scales with the number of dielectric lamina as has also been shown in work on electret based vibrational energy harvesters [5] and in water desalination [6].

The final, superhydrophobic layer, is produced by laser etching PMMA, and depositing a thin layer of PTFE on the resulting roughness topology to increase the Young's angle. The surface charge density can be modulated during the experiment using the high-voltage DC-DC converter, which can re-polarize the dielectric substrate by means of an embedded electrode, and the resulting bound charge partially or fully neutralizes the electric field due to the surface ions deposited by corona charging of the electret. The high-voltage system is armed manually before the drop and is automatically safed by a high-voltage power switching relay, which switches the load across a  $100\text{ k}\Omega$  resistor when triggered by the resumption of 1-g conditions. The safing is set by an accelerometer pin-interrupt triggered microcontroller command. The rig with a mounted experiment is shown in Figure 3.

Contact angles of distilled water on the electret  $\sim 150^\circ$ . The hysteresis of the contact angle (the difference between the advancing and receding contact angles) is estimated from the roll-off angle using the method of Furmidge, 1962 [7], and is found to be approximately  $20^\circ$  when the surface is uncharged. The innate Young's angle hydrophobicity of the PTFE layer is enhanced by the underlying roughness length scales of the surface. We use a laser-etched pillar geometry with pillar heights  $\sim 680\text{ }\mu\text{m}$ , widths  $\sim 70\text{ }\mu\text{m}$  and pitch  $\sim 100\text{ }\mu\text{m}$ . An SEM image of the pillar geometry is shown in Figure 5.

Surface potentials were measured on the superhydrophobic surface using a *Simco-Ion* FMX-004 electrostatic fieldmeter and the method for determination of surface charge density for low conductivity polymers described in Davies, 1967 [8]. This measurement was done with the superhydrophobic surfaces connected by a conductive ground plane by conductive tape, far away from the presence

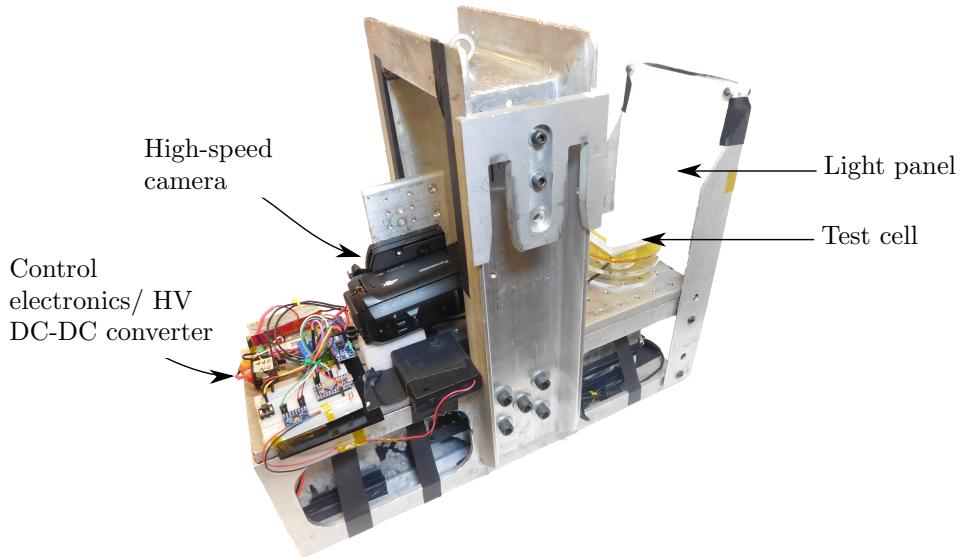


Figure 3: The droplet electro-bounce experiment hardware mounted on a drop tower rig.

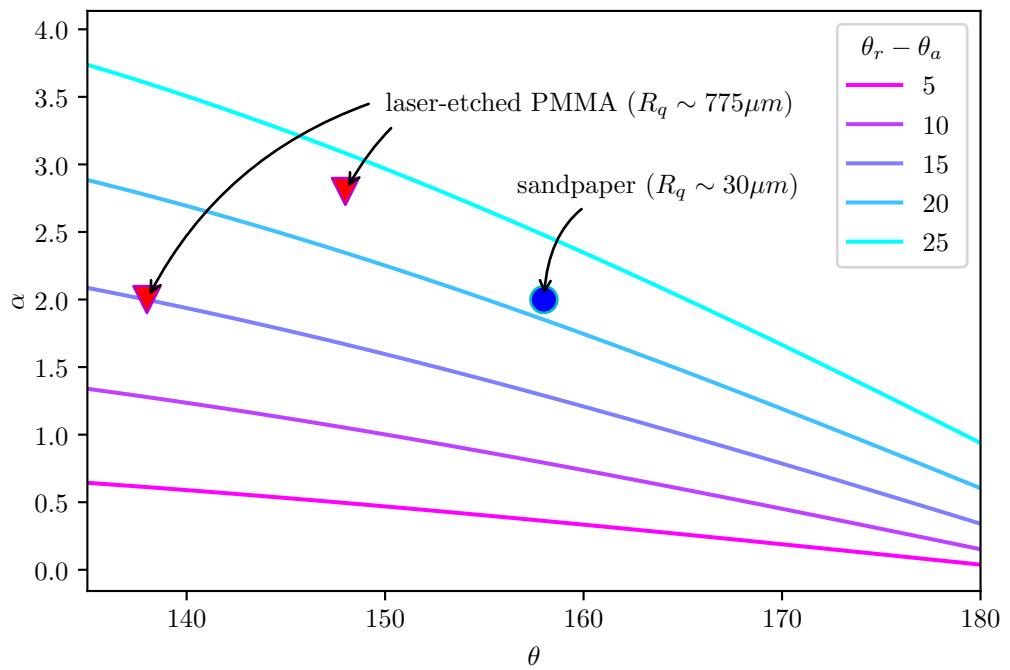


Figure 4: Hysteresis of the contact line plotted as a function of static contact angle and roll-off angle for 1 mL droplets.

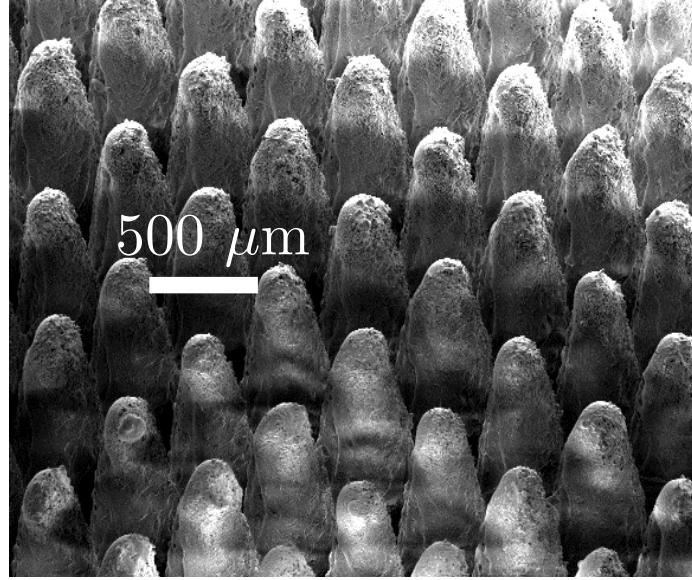


Figure 5: SEM image of the superhydrophobic surface.

of other conductors. An ideal approach to determining surface charge on a dielectric surface is to screen perturbing effects of external electric fields. This is partly accomplished by grounding the fieldmeter, and by placing the dielectric sample on a grounded conductive plate backing. In this case the surface charge density is determined from

$$\sigma = \frac{\varphi_s \kappa}{l},$$

where  $\varphi_s$  is the surface potential read by the fieldmeter, and where  $l$  and  $\kappa$  are the thickness of the dielectric surface, and the absolute permitivity respectively. The measured surface voltage is a function of position away from the charged dielectric. In most cases this function is relatively constant at a distance about 1-2 cm away from the surface (there is some measurement error in surface voltage due to small mispositioning of the electrostatic fieldmeter, say by  $\pm 1$  mm). The relative dielectric constant of the PMMA sheet was measured by using a 65 mm  $\square$  polished aluminum parallel plate capacitor with  $C = \epsilon \epsilon_0 \frac{A}{l}$  where  $C$  is the capacitance, and  $A$  is the sheet area. Measuring the capacitance with 3 sample thicknesses using a GenRad 1657 RLC Digibridge, we found the relative

permittivity to be  $\epsilon = 3.5 \pm 0.4$ .

A further consideration is the possibility of the change in total charge during a typical experimental timescale. If we consider the drop rig to be a ground (which seems reasonable given that the rig is isolated from true ground, but is at some reference voltage with respect to the surface charges on the dielectric, it also has an abundance of free charge carriers, that is, if it is conductive), then there will be both bulk and surface decay of the charge on the dielectric. The evolution of the charge can in some cases be approximated by

$$\sigma = \sigma_0 e^{\frac{-t}{\epsilon\rho}},$$

where  $\sigma_0$  is the initial surface charge density, and  $\rho$  is the bulk resistivity (which can also be reframed in terms of conductivity by  $\rho = 1/\gamma$ , where  $\gamma$  is the conductivity). For an example case of a surface with an initial surface charge density  $\sigma = 2.4 \times (10^{-6}) \text{ C/m}^2$ , relative permittivity  $\epsilon = 3.5$  and bulk resistivity  $\rho = 1.6 \times (10^{16}) \Omega\text{cm}$  such as with the case of 2.8 mm PMMA sheet, then the time constant  $\tau = \kappa\rho$  is approximately 5000 s, which is a great deal longer than the typical time period for of a drop tower experiment. We measured the charge decay with a calibrated probe at periodic intervals. There are several charge decay mechanisms: internal ones, such as Ohmic resistance, and external ones such as compensation by environmental ionic species. The relative magnitudes of these charge transport mechanisms, and therefore the stability of the electret varies drastically depending on its initial surface potential, material properties, environment, and charging method. In the case of unshielded electrets compensation by atmospheric ions is significant [9]. Because environmental convection will tend to maintain a gradient of these ions, sealing an electret in a container from the atmosphere will effectively halt this decay mechanism. Atmospheric humidity and water droplet condensation also significantly increase charge decay (presumably by reducing the surface resistance) [10]. Examples of this decay can be seen in Figure 6 for differing numbers of dielectric lamina used

in this experiment. In looking at the trends in charge decay for our electrets we notice firstly, that the decay does not appear to be exponential, as in the model described above. Secondly, we plainly see a cross-over effect in the decay of the surface potential in our electret samples, whereby the samples charged to higher initial surface potentials decayed faster and reached the lower overall final potentials. This is a well known effect in polyethylenes charged by corona [11].

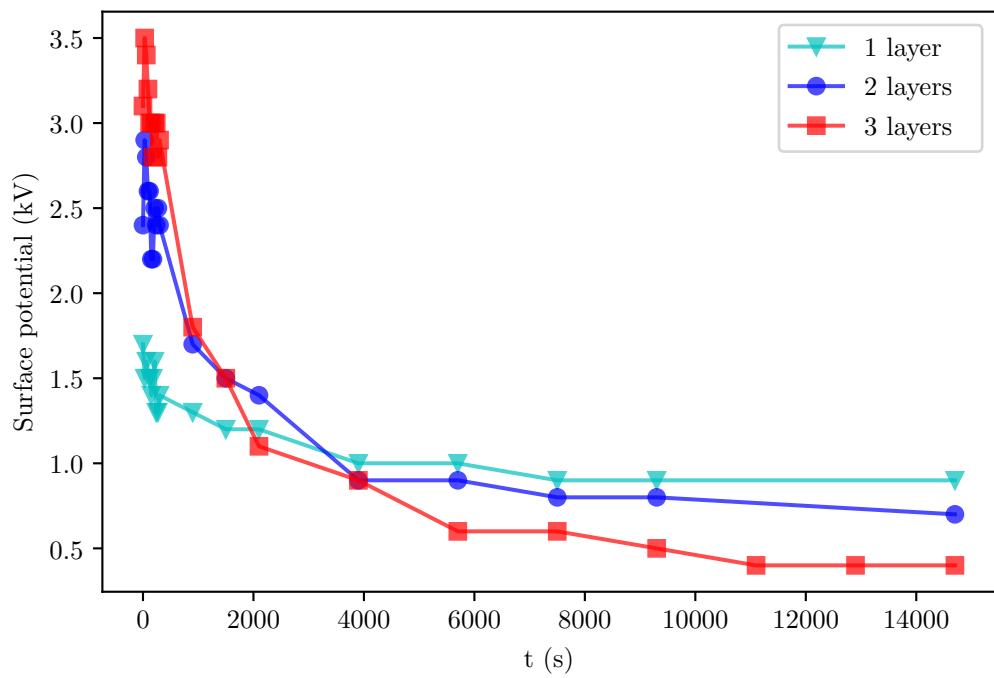


Figure 6: Charge decay in the dielectric laminates for differing numbers of layers.

### 3 Data Munging

Digitization of droplet trajectories requires several steps of post-processing. Video is first decomposed into a sequence of still images. Trajectories are captured using the particle tracking module in *Fiji* [12], a derivative of the popular *ImageJ* [13] package for scientific image analysis. The series is stabilized to remove the effect of drop transients from the kinematic data [14]. The series of still images is cropped, and the background (that is, the low-entropy pixels) of the series is removed using a builtin “rolling ball” algorithm. Each still is then split into its constituent RGB maps. In this case the green channel images contained the most information, so these were then globally thresholded using the Triangle algorithm to recover a map of the pixels corresponding to the droplet’s approximate position in the original still. Ellipses are fitted to the pixel map stepping through the time series to determine the positions of the center of mass, and the semi-major and minor axes of the droplets during the drop. Finally, a perspective correction is applied to the center-of-mass positions. A Python data class was made to object orient experimental metadata, as opposed to using HDF5 data format or some form of structured array. The code used in this work, as well as the raw data, and the make files for this thesis are archived on the open-source project portal Github [15].

### 4 Parameter Estimation

Using various scaling arguments we have gleaned from our simple model a series of non-dimensional numbers characteristic of droplet bounce apoapses and times of flight, but these dimensionless groups depend on a set of physical parameters. Unfortunately not all of these parameters are physically practical to accurately measure by experiment. Droplet free charge  $q$ , in particular, could in principle be directly measured by collecting the charged drops in a faraday cup under low-g and measuring the change in capacitance of the cup using a very high input-resistance electrometer, but this is a problematic experiment to set up

in a drop tower from a practical standpoint. The other state variables we can directly measure by experiment with varying levels of accuracy. To measure the charge,  $q$ , we instead turn to parameter estimation techniques. Our work flow to identify this parameter is as follows:

1. Experimentally vary  $V_d$ ,  $\sigma$  and capture droplet trajectories using a high-speed camera.
2. Digitize droplet trajectories by using automatic tracking of ellipse-fitted centroids on the thresholded video.
3. Slice droplet trajectories by their bounce minima, and apply a smoothing filter.
4. Extract the droplet charge (and other experimental parameters) by maximizing the log-likelihood of the data given the dynamical model and parameters, by varying the parameter vector using a direct search optimization.

Mathematically we state that we find the parameters  $\mathbf{x}$  that solve the inverse problem  $G(\mathbf{x}) = \mathbf{d}$ , using a direct search method (*Nelder-Mead*).

$$\min \chi^2 = \min \sum_{i=1}^n \frac{(y_d(t)_i - y_G(t, \mathbf{x})_i)^2}{y_G(t, \mathbf{x})_i}$$

$$\mathbf{x} = \begin{cases} q \\ V_d \\ \sigma \end{cases} \quad \text{subject to constraints } g = \begin{cases} V_d \pm u_{exp} \\ \sigma \pm u_{exp} \\ y_0 \pm u_{exp} \\ t_0 \pm u_{exp} \end{cases}$$

where  $y_G(t, \mathbf{x})$  is a numerical solution of the equation of motion

$$my'' = \frac{1}{2}\rho C_D A dy'^2 + qE(y) + Kq^2y^{-2}$$

## 4.1 Inverse Problems

Suppose we have a model  $G(\mathbf{x})$ , with a vector of parameters  $\mathbf{x}$ , and set of (noiseless) observations  $\mathbf{d}$ , the we naturally expect there to exist a relationship

$$G(\mathbf{x}) = \mathbf{d},$$

where the operator  $G$  might be an ODE. Suppose the model  $G(\mathbf{x})$  is the ODE

$$y'(t) = f(t, \mathbf{y}; \mathbf{x}), \quad \mathbf{y} \in \mathbb{R}^n,$$

and a collection of  $n$  measurements of experimental data

$$\mathbf{d} = (t_1, \mathbf{y}_1), (t_2, \mathbf{y}_2), \dots, (t_k, \mathbf{y}_k).$$

The process of fitting a function, defined by a collection of parameters, to a data set is called the discrete inverse, or parameter estimation problem (as opposed to the *forward problem* to find  $\mathbf{d}$  given  $\mathbf{x}$  and  $G(\mathbf{x})$ ). This is a familiar procedure when the determination of model parameters is done using linear or polynomial regression. However there are approaches even to fitting an arbitrary function to a noisy and sparse dataset. In this work we use the conventional Maximum Likelihood Estimate (MLE) method to identify the model parameters.

Using MLE we don't ask the question: "what is the probability that my set of model parameters is correct?" (because the probability is very nearly zero!), but rather "given my set of model parameters, what is the probability that this data set occurred (what is the likelihood of the parameters given the data)?". Bayes' Theorem holds that

$$\text{prob}(X|D, I) = \frac{\text{prob}(D|X, I) \times \text{prob}(X|I)}{\text{prob}(D|I)}$$

where  $D$  are our observations (dataset),  $X$  is our vector of parameters, and  $I$  is general background information about the problem including our mathematical

model (for instance the ODE above), and

$$\begin{aligned}
\text{prob}(X|D, I) &\quad \text{posterior probability density function,} \\
\text{prob}(D|X, I) &\quad \text{likelihood function,} \\
\text{prob}(X|I) &\quad \text{prior probability density function,} \\
\text{prob}(D|I) &\quad \text{evidence.}
\end{aligned}$$

The posterior probability density function (PDF)  $\text{prob}(X|D, I)$ , is ultimately what we want to estimate, the prior PDF  $\text{prob}(X|I)$ , reflects our knowledge of the system, and the evidence  $\text{prob}(D|I)$ , is the likelihood of the data based on our knowledge. We also note that since it only makes sense to compare the conditional PDF's for the same data, we can ignore the denominator (that is, the evidence). We further note that the prior  $\text{prob}(X|I)$ , is fixed before our observations and so can be treated as invariant to our problem. We can therefore infer that  $\text{prob}(X|D, I) \propto \text{prob}(D|X, I)$ . The MLE for the model parameters  $\mathbf{x}_0$ , then is given by the maximum of the posterior PDF, which is equivalent to the solution of the ODE given the the parameters  $\mathbf{x}$ , that produces the highest probability of the observed data. Since the likelihood  $\mathcal{L}(\mathbf{x}) = \prod_i^n \mathcal{P}_i$ , and the probability  $\mathcal{P}$ , of any single observation is less than one, then the total likelihood which is the product of a large number of probabilities tends to be vanishingly small. The more well behaved log-likelihood is given by

$$\mathcal{M} = \ln(\mathcal{L}) = \ln(\text{prob}(D|X, I)) = \text{const} - \frac{\chi^2}{2}$$

where

$$\chi^2 = \sum_{i=1}^n \frac{(y_{di} - y_{Gi})^2}{\sigma_{di}^2}$$

is the  $\chi^2$  goodness-of-fit,  $y_d = y_d(t)$  is a observation of droplet position at a point in time, and  $y_G = y_G(t, \mathbf{x})$  is the droplet position predicted by the solution to the equation of motion at time  $t$ , and  $\sigma$  is the standard error of the position measurement. If the number of data points  $n$ , is small was can use the

Poisson error  $\sigma_d^2 = y_G$ . The optimal parameter set is the one with the highest probability of observing the data (the maximum of the posterior PDF) and can be determined by maximizing the log-likelihood  $\mathcal{M}$  (or minimizing  $\chi^2$ ) of the data  $\mathbf{d}$  with respect to the parameter set  $\mathbf{x}$ . Thus parameter estimation is a variety of optimization problem.

## 4.2 Smoothing

All optimization methods, whether explicitly or implicitly, follow gradients towards an optimum. In a parameter estimation problem, if we approximate these gradients by finite differences, then the noise manifests itself as amplification of the roughness in the hyper-response surface. Gradient based optimizers do poorly in these situations because they tend to converge to local minima. While so called gradient free algorithms offer an improvement in this regard, speed of convergence and the quality of the MLE is improved by smoothing the objective function. This is equivalent to smoothing the underlying dataset.

Our choice of smoothing approach depends principally on the nature of the errors in the dataset. The sources of error include misalignment of the camera, error in the fiduciary length scale, perspective due to objects (subject or reference scale) being out of the photographic plane, and various errors arising in the digitization process (including the difference between the thresholded ellipse fitted centroid and the true centroid of the non-ellipse drop centroid). Some of these errors are systematic in origin and introduce consistent biases into the data (e.g. coherent spectral sources, rather than truly stochastic noise). Data smoothing does little to help systematic errors in that they are usually of lower frequency than the signal. Random errors, by contrast, are assumed to have a Gaussian distribution (by the central limit theorem), and are independent of the signal (which inherently results from a deterministic process).

We experimented with a variety of filters implemented in the `scipy.signal` Python *SciPy* [16] module on a representative set of trajectory data; these methods include 1D Gaussian convolution, Wiener, Butterworth, and Savitsky-Golay

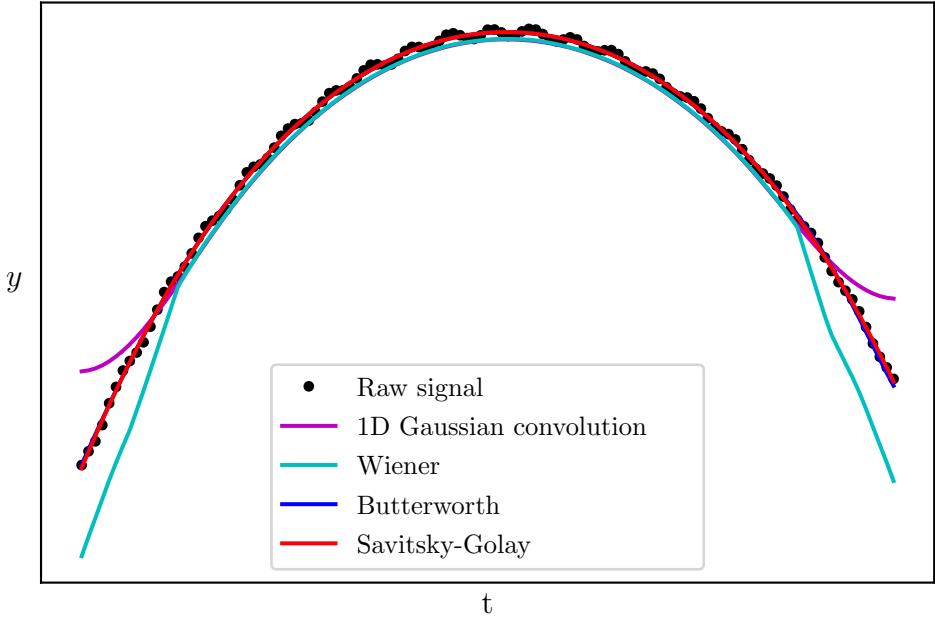


Figure 7: The underlying signal is ‘noisy’, due partially to deterministic errors in determining the center of mass position. These deterministic errors are largely due to droplet oscillations, especially the rapidly damped higher harmonics which do not have azimuthal symmetry. There is also Gaussian error in the ellipse fitting due to thresholding and noise in the video itself. We see that 1D Gaussian convolution and Wiener filters suffer from significant end effects. At this scale Butterworth and Savitsky-Golay filters are nearly indistinguishable.

filters. Qualitatively comparing these smoothing methods (by hand tuning filter orders and window sizes) we find that we loose too many data points in the smoothing process, large amplitudes are overly smoothed by repeated filtering passes, or there are significant end effects for most of these methods. A comparison of these smoothing approaches on a representative trajectory data set is shown in Figure 7.

The Savitsky-Golay, and Butterworth filters both produce fairly smooth derivatives as can be seen in Figure 8; but the small window-size needed for Butterworth filter tends to also produce a noticeable end effect. The Savitsky-Golay filter essentially uses a moving-window based on local least-squares polynomial approximations. It was shown that fitting a polynomial to a set of input samples and then evaluating the resulting polynomial at a single point within

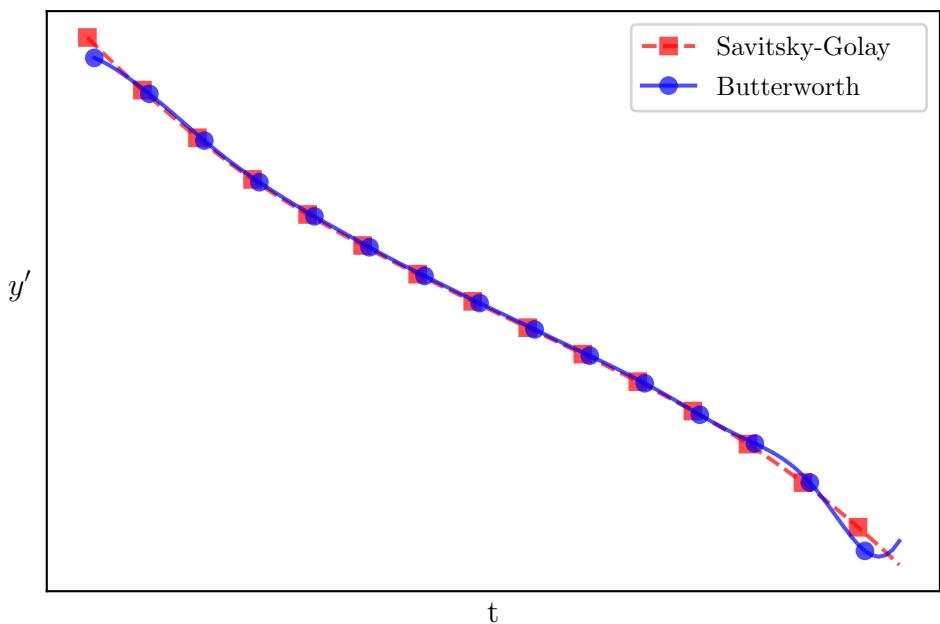


Figure 8: Comparing the first derivatives of the Butterworth and Savitsky-Golay filters we see that the Butterworth filter also suffers from a slight end effect. This implies that the optimization will find a different optima of the likelihood depending on which type of filter is used.

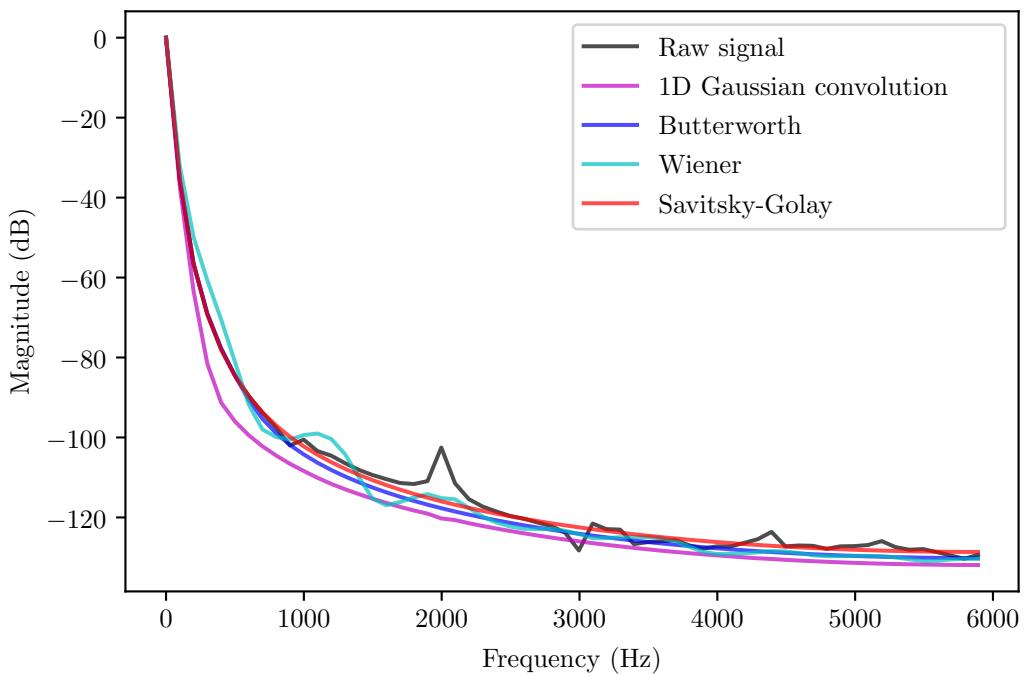


Figure 9: The power spectra has a peak at 1 Hz, which is the droplet trajectory parabola itself, and smaller peaks in the kHz range corresponding to various noise frequencies. The Savitsky-Golay and Butterworth filters seem to have the least distortion of the power spectra of the true signal. Both also do a good job of attenuating the noise at the 2 kHz peak.

the approximation interval is equivalent to discrete convolution with a fixed impulse response [17]. A beneficial property of this kind of low-pass filter is their tendency to maintain waveform amplitudes, and so they are attractive in applications having noisy signals with sharply pointed waveforms such as ultrasound or synthetic aperture radar [18]. Because Savitsky-Golay is a Finite Impulse Response (FIR) filter it requires data points to be equally spaced; to accommodate this requirement we interpolate points between the small gaps which sometime occur in the tracking results from image analysis. We use a moving window size slightly smaller than the length of the current bounce in a single drop jump data set. The windows are piecewise defined by partitioning the data set into a series of individual bounces (the dataset is sliced at minima identified after an initial rough smoothing pass, using the `scipy.signal.argrelextrema()` function). The Savitsky-Golay polynomial order is 4. To understand how these filters differ it is useful to look at their frequency response. In Fourier space, convolution becomes a multiplication, and we can understand what a filter does by looking at which frequencies it lets pass through. We can do this using a Discrete Fourier Transform (though it is worth noting that our signal is not truly periodic). The power spectra for these same data are shown in Figure 9.

### 4.3 Optimization

Most generally a constrained optimization problem is stated as

$$\begin{aligned} \text{minimize: } & f(\mathbf{x}) && \text{objective function} \\ \text{subject to: } & g_j(\mathbf{x}) \leq 0 && \text{inequality constraints} \\ & h_k(\mathbf{x}) = 0 && \text{equality constraints} \end{aligned}$$

$$\text{where } \mathbf{x} = \begin{cases} x_1 \\ x_2 \\ \vdots \\ x_n \end{cases} \quad \text{design variables}$$

Mathematical optimization is the problem of finding minima of a function  $f$ . In this context the function is called the cost, or objective function. The field of mathematical optimization is as old as calculus itself, and the number of particular optimization techniques is correspondingly myriad; particular techniques lend themselves well to particular types of optimization problems. The minima of the objective function  $f$  is sought on a domain  $A$  specified by the constraints of the problem; this domain is usually called the feasible region. Minima of objective function  $f : A \rightarrow \mathbb{R}^m$  are called feasible solutions. If the function  $f$  is convex the feasible solution is the global minimum, otherwise additional local minima exist. The scale of the optimization problem is set ultimately by dimensionality of the objective function. Functions may not always be smooth in the sense of having continuous derivatives, and this is problematic in that optimization methods fundamentally rely on gradients of the objective function. Problems with anisotropic objective functions where there is strong covariance between the parameters, and the gradient vector generally to differ significantly from the Newton direction  $(-\mathbf{H}^{-1}f'(\mathbf{x})^T$ , where  $\mathbf{H}$  is the Hessian matrix) are considered ill-conditioned. Ill-conditioned problems gradient based deterministic search tend to converge slowly as they take a zigzagging path determined by the local value of the gradient rather than following the Newton-direction vector which towards the minimum. Numerical optimization may deal with black box functions (where we do not have an explicit mathematical expression of the function we are optimizing). Black box problems are challenging because we do not have access to analytic gradients of the objective function, and approximating them by finite-differences is slow and noisy. In general, noisy, black box, non-linear, non-quadratic, non-convex, constrained, ill-conditioned, high-

dimensional objective functions are problematic to optimize. Unfortunately, problems of this type are the essence of the parameter estimation, which often leads to its characterization as an ‘art’ rather than a precise science (though we submit that it is a dark art).

The equation of motion behaves stiffly due to the large disparity in Coulombic, image charge length scales. We integrate it numerically using the `odeint` *Scipy* module. This is a shake-and-bake Python wrapper for the venerable 1982 *netlib ODEPACK* library double-precision `lsoda` (Livermore Solver for Ordinary Differential equations with Automatic method switching for stiff and non-stiff problems) integrator [19]. The function switches between Adams (nonstiff) and Backwards Differentiation Formulas (BDF, stiff) according to the dynamic value of a set of stiffness eigenvalues.

Our specific optimization problem is non-convex, mixed discrete-continuous black-box (noisy), and highly ill-conditioned which is essentially the worst case scenario for an optimization problem. The ill-conditioning arises due to the strong covariance between several of the model parameters (particularly  $q = q(V_d, E_0)$ ). The non-convexity of the problem implies that there are many local minima of the objective function. While in principle a gradient-based optimizer (for instance using the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS) [20]) could be used by using finite-differences to obtain approximate gradients of the  $\chi^2$  objective function, in practice doing so is extremely problematic because the noise-to-signal ratio of the objective function scales like  $\mathcal{O}(f)$  for  $\frac{df}{dt}$  and  $\mathcal{O}(f^2)$  for  $\frac{d^2f}{dt^2}$  which will tend to cause convergence to a local minima which is only an artifact of the likelihood response surface [21]. As a further practical matter, given the relatively expensive function-calls (which requires solving a stiff, non-linear ODE) gradient-free approaches tend to offer better performance regardless [22].

We use a gradient-free, direct-search approach: Nelder-Mead [23] implemented in `scipy.optimize`. Nelder-Mead is robust to noise (relatively speaking), and is thrifty with our extremely expensive function-calls. *Nelder-Mead*,

sometimes called simplex-search or downhill-simplex, is a heuristic search method, with no guarantee of optimal solutions, but is well-established and widely used despite that. *Nelder-Mead* is based on the concept of a  $N$ -simplex, which generalizes a triangle into higher dimensions as a polytrope of  $N + 1$  vertices in  $N$  dimensions. It uses only-function calls and expands or contracts the simplex according to the function values at its vertices in a way visually reminiscent (in  $\mathbb{R}^3$ ) of the oscillations of the jumping droplets themselves (in fact *Nelder-Mead* is sometimes also called the “amoeba method”). Very little is known about the convergence properties of the *Nelder-Mead* algorithm in its classical form for non-smooth objective functions [24], except that in general it doesn’t satisfy the properties required for convergence by other direct search algorithms: that the simplex remains uniformly non-degenerate, and that some form of “sufficient” descent condition for function values at the vertices is required at each iteration. Scaling can help solve convergence problems and improve numerical stability. We precondition the optimization problem by minimizing  $\ln(\chi^2)$ , and using a naive scaling (scaling variables such that their magnitudes  $\sim \mathcal{O}(1)$ ) of our constraints by their initial guesses. Here is goal is to make the problem equally sensitive to steps in any direction. *Nelder-Mead* is not a global optimizer, though there are variants which use sequential local searches with probabilistic restarts to achieve globality. However global optimization usually comes at a tremendous computational cost. However, *Nelder-Mead* behaves less locally than many gradient-based approaches. The convergence history of the parameter MLE using *Nelder-Mead* for a single drop jump experiment is shown in Figure 10. Results of the parameter estimation are shown in Figures 11, 12.

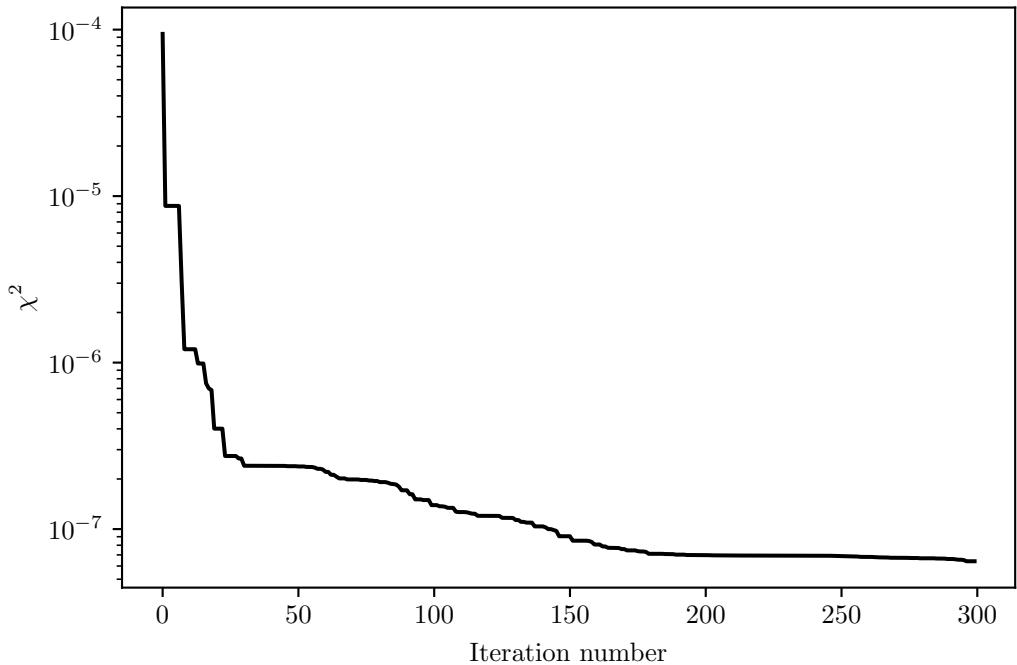


Figure 10: As is typical with *Nelder-Mead* much of the improvement in  $\chi^2$  is realized in the first few iterations. Overall the rate of convergence is sub-linear, which is to be expected for non-linear constrained problems using an heuristic algorithm.

#### 4.4 Identifiability

That we are capable of fitting any arbitrary model to a dataset given sufficient degrees of freedom in our parameters is admittedly a disconcerting issue, begging the question “given the structure of the model is it possible to uniquely estimate the unknown parameters?” This question is called the problem of identifiability. However, some of the inverse model parameters are constrained by our experimental observations of them and their associated measurement uncertainties. This, we hope, makes the specter of an over fitted model less frightening, but does convert our unconstrained optimization problem to an constrained one which raises special difficulties of its own, which we discuss below.

The *Nelder-Mead* direct search method cannot be used with explicitly constrained problems. However, there are various implicit approaches to (approx-

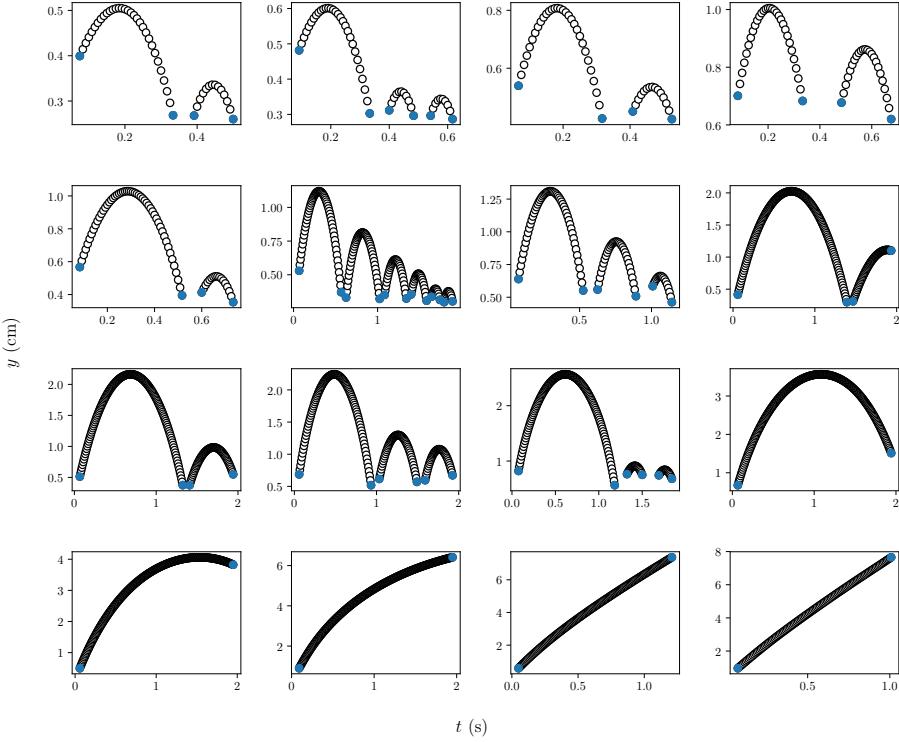


Figure 11: A series of filtered droplet trajectories arranged by increasing apoapse. The blue dots represent either the beginning and end of the experiment, or points at which the droplet is either coming into, or leaving contact with the surface.

imately) solving general constrained problems using unconstrained algorithms. Generally, this is achieved by domain transformations or the use of penalty functions. By the addition of a penalty function which depends in some way on the values of the constraints to the objective function, we minimize a pseudo-objective function where the infeasibility of the constraints is minimized simultaneously to the objective function.

There are various penalty function schemes. We use an Exterior Penalty Function as a simple way of converting the constrained problem into an unconstrained one. These are especially useful in cases where the constraints are not “hard” in the sense that they need to be satisfied precisely. General penalty functions, which are sequential unconstrained minimization techniques, reformulate the general constrained problem as the pseudo-objective

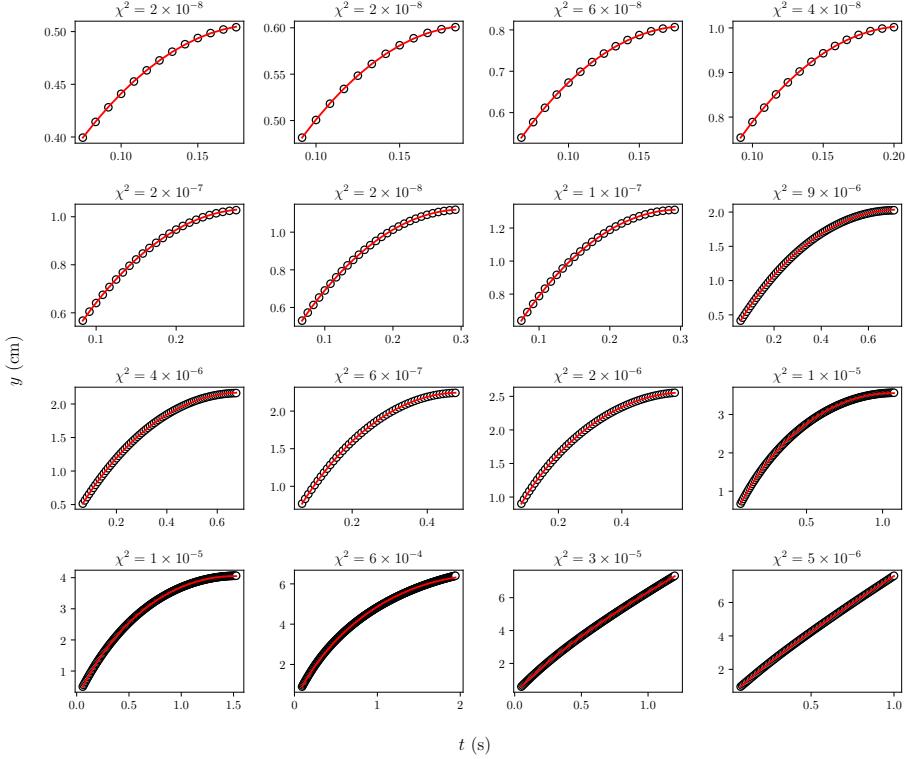


Figure 12: A series of droplet trajectories showing the results of the parameter estimation. The trajectories are shown only up to the apoapse of the first bounce. The red lines show the ODE solution with given the MLE parameter vector.  $\chi^2$  goodness-of-fit varies between  $1 \times 10^{-5}$  and  $1 \times 10^{-8}$  with the better fit occurring typically for the droplets with the lowest apoapses.

function given by

$$\phi(\mathbf{x}, r_p) = F(\mathbf{x}) + r_p P(\mathbf{x})$$

where  $P(\mathbf{x})$  the penalty function, is given by

$$P(\mathbf{x}) = \sum_{j=1}^m \{\max[0, g_j(\mathbf{x})]\}^2 + \sum_{k=1}^l [h_k(\mathbf{x})]^2. \quad (1)$$

We see from Equation 1 that there is no penalty if the constraints  $g_j(\mathbf{x})$ , and  $h_k(\mathbf{x})$  are satisfied.

The Exterior Penalty Function specifically (and all Penalty Function ap-

proaches in general) do have several drawbacks. Namely these include the possibility of the objective function being undefined outside of the set of feasible solutions. Additionally, by naively “encouraging” feasibility of the solution using large values of the penalty parameter,  $r_p$ , we will tend to ill-condition the unconstrained formulation of the problem (though in our implementation the preconditioning tends to make the pseudo-objective function less and less sensitive to the constraints as the likelihood is approaches a maximum). We use the measured values of  $u_0$ ,  $V_d$ , and  $E_0$ , and the informed guess  $q \approx kV_d^{2/3}E_0$  where  $k$  is a constant  $k \approx 1^{-11}$  as an initial guess for the parameter vector  $\mathbf{x}$ . We stop the optimization after 300 iterations rather than waiting for convergence.

We’re interested in the variance and co-variance as a means to determine the quality if the parameter estimate. The  $(i,j)$ -th element of the matrix  $\sigma(\mathbf{x}, \mathbf{y})$  is equal to the covariance  $\text{cov}(X_i, Y_j)$  between the  $i$ -th scalar component of  $\mathbf{x}$  and the  $j$ -th scalar component of  $\mathbf{y}$ . Here the concept of error bars in linear correlation associated with a covariance matrix are not suitable. We might try to generalize the idea of confidence intervals to a multidimensional space, but it is likely difficult to describe the surface of the (smallest) hyper-volume containing 90% of the probability with just a few numbers. The situation becomes worse if the probability density function has several maxima. However, there is another approach to find the covariance matrix [ref]

$$[\sigma^2]_{ij} = - \left[ (\nabla^2 \mathcal{L})^{-1} \right]_{ij} = 2 [\nabla^2 (\chi^2)]_{ij}^{-1} = - [H^{-1}]_{ij}$$

where  $H$  refers to the Hessian matrix and  $[\sigma^2]_{ij}$  is the covariance matrix  $C$ . The issue of identifiability is especially fraught for non-linear, black box type problems were it is difficult to explicitly evaluate the Hessian. The likelihood function (and thus the posterior probability density function) are defined completely by the optimal solution  $\mathbf{x}$  and the second derivative of  $\mathcal{L}$  at the maximum, which corresponds to the covariance matrix  $C$ . The standard errors (marginal variances) are the square roots of the diagonal of the covariance matrix. The

Hessian matrix must be negative definite for  $\mathcal{L}$  to have a maximum at  $\mathbf{x}_0$ . We obtain an approximate Hessian using the approach suggested in Nelder's original paper, by fitting quadratics to the simplex vertices and midpoints [23][25]. We can use the condition number

$$\text{cond}(H) = \frac{\max[\text{eig}(H)]}{\min[\text{eig}(H)]}$$

to make a qualitative assessment of the stability of the problem. We find typical condition numbers  $\sim \mathcal{O}(10^{27})$ . In fact, in many cases our Hessian is not negative definite, or invertible at the minima  $\mathbf{x}_0$ , which implies negative values of the variances (which is impossible). These issues indicate the problem is extremely ill-conditioned near the minima  $\mathbf{x}_0$ . Given that the parameter estimates are highly colinear, it is unsurprising that the Hessian matrix is near-singular and its inversion is either impossible or involves significant numerical error. Posterior verification of the results is crucial to bound the identifiability of the parameter estimation problem, but it remains as future work. Several sources suggest sensitivity analysis of the parameter estimates by generation of synthetic Monte Carlo data sets as a means of establishing identifiability [26][27].

## References

<sup>1</sup>*Electrets — gerhard m. sessler — springer ()*.

<sup>2</sup>O. Heaviside, *Electrical papers*, Vol. 2 (Cambridge University Press, 2011).

<sup>3</sup>T. Wu, Y. Suzuki, and N. Kasagi, “Low-voltage droplet manipulation using liquid dielectrophoresis on electret”, *Journal of Micromechanics and Microengineering* **20**, 085043 (2010).

<sup>4</sup>J. Van Turnhout, *Thermally stimulated discharge of polymer electrets: a study on nonisothermal dielectric relaxation phenomena* (Elsevier Science & Technology, 1975).

- <sup>5</sup>Y. Wada, H. Oguchi, M. Hara, H. Asanuma, and H. Kuwano, “Stacking electrets for electrostatic vibration energy harvesters”, in Digest tech. Power-MEMS 2012 workshop (2012), pp. 504–507.
- <sup>6</sup>H. Ni, R. C. Amme, and Y. Jin, “Desalination by electret technology”, Desalination **174**, 237–245 (2005).
- <sup>7</sup>C. G. L. Furmidge, “Studies at phase interfaces. i. the sliding of liquid drops on solid surfaces and a theory for spray retention”, Journal of Colloid Science **17**, 309–324 (1962).
- <sup>8</sup>D. K. Davies, “The examination of the electrical properties of insulators by surface charge measurement”, Journal of Scientific Instruments **44**, 521 (1967).
- <sup>9</sup>M. M. Perlman, *Electrets, charge storage, and transport in dielectrics*, Vol. 2 (Dielectrics and Insulation Division, Electrochemical Society, 1973).
- <sup>10</sup>H. T. M. Haenen, “The characteristic decay with time of surface charges on dielectrics”, Journal of Electrostatics **1**, 173–185 (1975).
- <sup>11</sup>G. F. L. Ferreira and M. T. Figueiredo, “Corona charging of electrets: models and results”, IEEE Transactions on Electrical Insulation **27**, 719–738 (1992).
- <sup>12</sup>J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: an open-source platform for biological-image analysis”, Nature Methods **9**, 676–682 (2012).
- <sup>13</sup>C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “NIH image to ImageJ: 25 years of image analysis”, Nature Methods **9**, 671–675 (2012).
- <sup>14</sup>K. Li, *The image stabilizer plugin for ImageJ* (Feb. 2008).
- <sup>15</sup>E. Schmidt, *Droplet-electro-bounce: project files for droplet electro-bounce*, original-date: 2017-06-06T22:20:50Z, June 18, 2017.
- <sup>16</sup>T. E. Oliphant, “Python for scientific computing”, Computing in Science Engineering **9**, 10–20 (2007).

<sup>17</sup>A. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures.”, *Analytical Chemistry* **36**, 1627–1639 (1964).

<sup>18</sup>R. W. Schafer, “What is a savitzky-golay filter? [lecture notes]”, *IEEE Signal Processing Magazine* **28**, 111–117 (2011).

<sup>19</sup>A. C. Hindmarsh, “ODEPACK, a systematized collection of ODE solvers, RS stepleman et al.(eds.), north-holland, amsterdam,(vol. 1 of), pp. 55-64.”, *IMACS transactions on scientific computation* **1**, 55–64 (1983).

<sup>20</sup>*Numerical optimization — jorge nocedal — springer* ().

<sup>21</sup>G. A. Wood, “Data smoothing and differentiation procedures in biomechanics”, *Exercise and Sport Sciences Reviews* **10**, 308 (1982).

<sup>22</sup>T. Kolda, R. Lewis, and V. Torczon, “Optimization by direct search: new perspectives on some classical and modern methods”, *SIAM Review* **45**, 385–482 (2003).

<sup>23</sup>J. A. Nelder and R. Mead, “A simplex method for function minimization”, *The Computer Journal* **7**, 308–313 (1965).

<sup>24</sup>C. J. Price, I. D. Coope, and D. Byatt, “A convergent variant of the nelder-mead algorithm”, *Journal of Optimization Theory and Applications* **113**, 5–19 (2002).

<sup>25</sup>W. Spendley, G. R. Hext, and F. R. Himsorth, “Sequential application of simplex designs in optimisation and evolutionary operation”, *Technometrics* **4**, 441–461 (1962).

<sup>26</sup>P. R. Bevington, *Data reduction and error analysis for the physical sciences* (1969).

<sup>27</sup>K. Jaqaman and G. Danuser, “Linking data to models: data regression”, *Nature Reviews Molecular Cell Biology* **7**, 813 (2006).