Evaluating derivatives of experimental data using smoothing splines

Brenden P. Epps[1], Tadd T. Truscott[2], and Alexandra H. Techet[1]

[1]Department of Mechanical Engineering,
Massachusetts Institute of Technology, Cambridge, MA, USA
{bepps, ahtechet}@mit.edu

[2]Naval Undersea Warfare Center,
Newport, RI, USA
tadd.truscott@gmail.com

Abstract — *Instantaneous derivatives of high-precision experimental data can be evaluated by fitting the data with a smoothing spline. This paper presents a novel and robust method for choosing the best spline fit and, hence, the best prediction of the derivatives. Typically, a smoothing spline is fit by choosing the value of a smoothing parameter that controls the tradeoff between error to the data and roughness of the spline. However, this method can yield an unsatisfactory fit, because the roughness of the fitted spline is extremely sensitive to the choice of smoothing parameter. An alternate fitting method is to choose an error tolerance and to find the spline with the least roughness possible, given that the error must be less than or equal to this tolerance. In this paper, we systematically explore the relationship between error tolerance and the minimum possible roughness of smoothing splines. We find that there exists a critical error tolerance, corresponding to the spline that has the minimum error to the data possible, without also having roughness due to the noise in the data. Both an analytic example and an experimental example are presented.*

Keywords — *smoothing spline, data regression, noisy experimental data, derivative*

1 Introduction

Finding the rate of change of a measured quantity is a ubiquitous experimental task. The present work is motivated by the canonical physics problem of finding the *velocity* (rate of change of position) and *acceleration* (rate of change of velocity) of a sphere falling into a basin of water. Other examples can be found in many engineering disciplines. From fluid mechanics, consider computing a spatial $velocity\ gradient = \frac{d(velocity)}{d(position)}$ in order to determine the shear stress on a body, given experimental velocity field data. From solid mechanics, consider determining the $strain\ rate = \frac{d(strain)}{d(time)}$ of a material during a crash impact test, given strain data measured at several times during the experiment. From image processing, consider finding the *curvature* (i.e. second spatial derivative) of a feature found using standard edge detection algorithms on a digital image. In all these cases, experimentalists desire the derivative of a measured quantity.

This paper presents a method to determine derivatives of experimental data. Our method applies to data that is highly-resolved and has small experimental error (the need for these restrictions will be made apparent in Section 2.1). Consider a general set of experimental measurements

$$\tilde{y}_i = y(t_i) + \tilde{\epsilon}_i \tag{1}$$

made at times, t_i ($i = 1, \ldots, N$), where $y(t_i)$ is the true value of some smoothly-changing quantity and $\tilde{\epsilon}_i$ is the measurement error[1]. The goal of the present work is to examine experimental $\tilde{y}(t_i)$ data and to best approximate the true function it represents, as well as its first few derivatives

$$y(t),\ \frac{dy(t)}{dt},\ \frac{d^2y(t)}{dt^2},\ \frac{d^3y(t)}{dt^3}$$

Typically in experimental research, the true function is either unknown or too complex to be represented by a simple parameterized model (e.g. a single polynomial with unknown coefficients). In this case, the appropriate way to represent the unknown function is to fit the data with a *smoothing spline*. This spline does not require any knowledge about the true function (aside from assuming that it is piecewise continuous and smooth), and derivatives of this spline can be computed exactly.

A smoothing spline can be formed by a piecewise polynomial of degree n, with $n-1$ continuous derivatives at each break point. Typically, cubic ($n = 3$) or quintic ($n = 5$) polynomials are used. A particular spline, $s(t)$, can be characterized by its *error*

$$\breve{E}(s) \quad = \sum_{i=1}^{N} |\tilde{y}_i - s(t_i)|^2 \triangle t \tag{2}$$

and *roughness*, which is defined for cubic and quintic splines as follows

$$R_2(s) \quad = \int_{t_1}^{t_N} \left| \tfrac{d^2 s}{dt^2} \right|^2 dt \quad \text{(cubic spline)} \tag{3}$$

$$R_3(s) \quad = \int_{t_1}^{t_N} \left| \tfrac{d^3 s}{dt^3} \right|^2 dt \quad \text{(quintic spline)} \tag{4}$$

Further background on the mathematical formulation of smoothing splines and their application to measured data can be found in references [3, 12].

In the vast majority of the smoothing spline literature, researchers try to find the 'best' smoothing spline fit by minimizing the quantity

$$J(s) = p\breve{E}(s) + (1-p)R(s) \tag{5}$$

where the *smoothing parameter*, p, controls the amount of smoothing (e.g. [2], [5], [7], [10], [13], [14],). Note that p must be chosen a-priori. If you pick $p = 1$, then minimizing $J(s)$ requires minimizing $\breve{E}(s)$, which happens when the spline passes through every data point. If you pick $p = 0$, then minimizing $J(s)$ requires minimizing $R(s)$; roughness is zero for a cubic spline that is composed of linear segments (and zero for a quintic spline composed of quadratic segments). For any $p \in [0, 1]$, there exists a unique spline that minimizes $J(s)$ [3]. Let us call this problem of identifying the best p and minimizing $J(s)$ the 'de Boor formulation'.

[1] Assume that the time at which each measurement took place was measured exactly. Also note, the present discussion is not restricted to functions of time and can be applied to any univariate data.

Our approach to the spline fitting problem follows Reinsch [9]: We choose an *error tolerance*, E, and find the spline with the least roughness, given that the error must be less than or equal to this error tolerance.

$$\text{minimize } R(s), \quad \text{requiring } \breve{E}(s) \leq E \tag{6}$$

One can show that this roughness minimization problem, hereafter referred to as the 'Reinsch formulation', is equivalent to minimizing $J(s)$ in the above 'de Boor formulation' [3]. Note that $E = 0$ in the Reinsch formulation is equivalent to $p = 1$ in the de Boor formulation, and $E \to \infty$ in the Reinsch formulation is equivalent to $p = 0$ in the de Boor formulation. The problem now is to choose the 'best' error tolerance, E, for a given data set.

We propose that the solution of the Reinsch problem provides a relationship between minimum roughness and error tolerance, $R(E)$. One could evaluate (6) for several error tolerance values and compute the roughness of each resulting spline. In doing so, one would generate an *efficient frontier* of smoothing splines that are viable candidates for the best fit. Note that there is a one-to-one correspondence between E in (6) and p in (5), so for any given choice of E or p, one will find some spline on the efficient frontier. Therefore, a comparison of methods for choosing p (such as *generalized cross-validation* [2]) are out of the scope of the present work. It is important to note that the preceding discussion assumes that the breakpoints of the spline are at the data sites (i.e. the times). Methods exist to improve a spline fit by choosing alternate breakpoints [3], but this is also out of the scope of the present work. We propose that (for a given set of breakpoints) inspection of the efficient frontier is a general method for choosing the best spline fit.

Herein, we show that for high-resolution, high-precision data, the smoothing spline *roughness versus error tolerance efficient frontier* takes on a special form, which can be exploited to determine the best spline fit. In this case, there exists a critical error tolerance, E_{cr}, which corresponds to the 'best fit' spline. This 'best' spline fit is defined as the one that has the minimum error possible to the data, without being laden with roughness due to the noise in the data. For error tolerances greater than E_{cr}, a spline fit to noisy data will still be smooth, but it will have more error than the spline corresponding to that with error E_{cr}. For error tolerances less than E_{cr}, the minimum-roughness spline is still very rough, since it must follow very closely to the error-ridden data points. In this paper, we present a method for selecting the 'best' smoothing spline by identifying the critical error tolerance on the $R(E)$ frontier.

2 Analytic example

In this section, we consider an analytic spline fitting example, and we show that there exists an *efficient frontier* of spline roughness versus error tolerance. We then show how the shape of this frontier allows one to determine the best smoothing spline fit for a given data set.

Consider noisy "experimental data" constructed using the function $y(t) = e^{-t} \cdot \sin(t)$ and zero-mean normally-distributed "measurement error",

$$\tilde{y}(t_i) = e^{-t_i} \cdot \sin(t_i) + \mathcal{N}(0, \epsilon^2) \tag{7}$$

with $t_i = i \cdot \triangle t$ and $i = 1 \ldots N$. In figure 1a, we choose $\epsilon = 10^{-2}$, $\triangle t = 10^{-2}$, and $N = 10^3$.

By solving the 'Reinsch problem' (6) for several error tolerances, E, and then evaluating the roughness, R, of each output spline, one finds that there exists an *roughness versus error tolerance efficient frontier* for least-roughness smoothing splines. (For any given error tolerance, splines exist with more roughness than the one on the $R(E)$ frontier, but these are undesirable.) Two such frontiers are shown in figure 1b, one corresponding to quintic splines fit to the noisy $\epsilon = 10^{-2}$ data ('♦'), and one corresponding to quintic splines fit to the analytic $y(t_i)$ data ('•'). One striking feature of the $\epsilon = 10^{-2}$ frontier is the kink at $E = 1.3 \times 10^{-3}$.

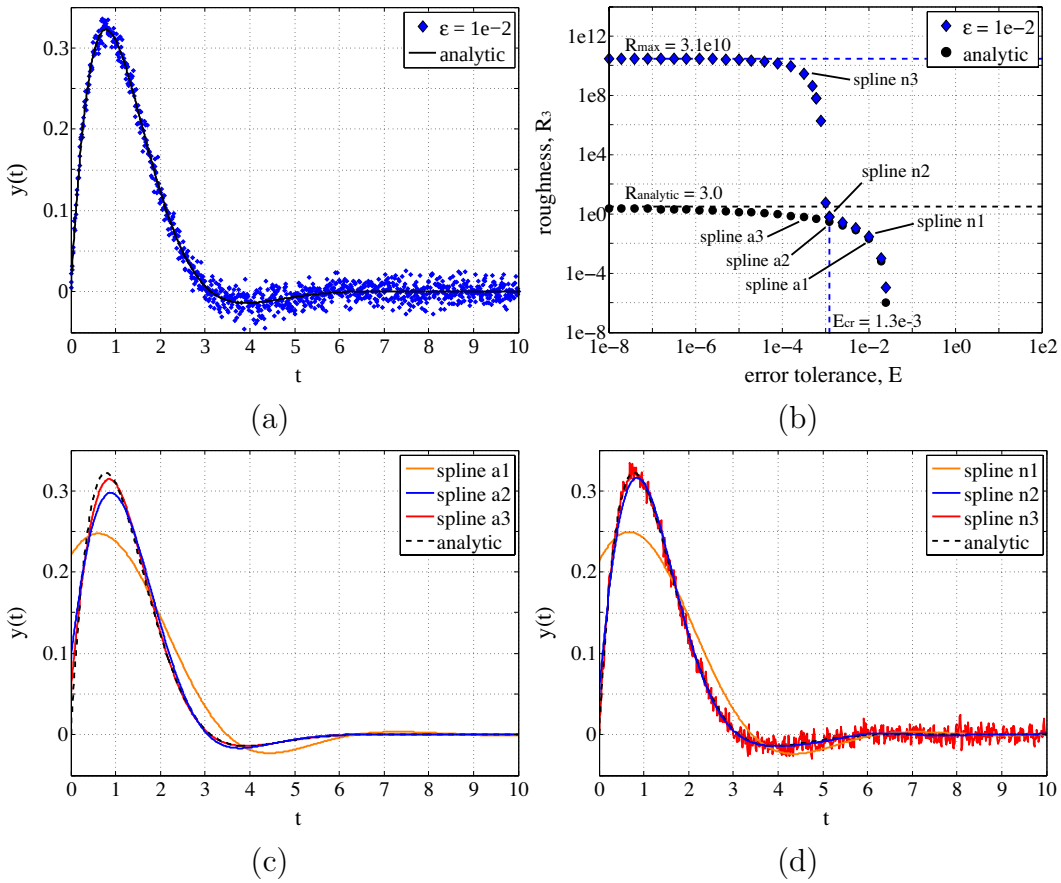

Figure 1: (a) Example analytic function $y(t) = e^{-t} \cdot \sin(t)$ and noisy 'data' $\tilde{y}(t) = y(t) + \mathcal{N}(0, \epsilon^2)$ with $\epsilon = 10^{-2}$. (b) Roughness, R_3, versus error tolerance, E, of quintic splines found by solving the Reinsch problem. Note: each point represents a particular spline fit. The roughnesses of the spline fits to the analytic data, '•', asymptotically reach the analytically-computed roughness of $y(t)$, whereas the roughnesses of the spline fits to the noisy data, '⧫', follow this trend for E larger than a *critical error tolerance*, E_{cr}, but increase several orders of magnitude for $E < E_{\text{cr}}$. (c) Selected splines fit to the analytic data. (d) Selected splines fit to the noisy data. Spline n2 is the fit with the smallest error tolerance that still mimics its corresponding spline fit to the analytic data.

We define the *critical error tolerance*, E_{cr} ($= 1.3 \times 10^{-3}$ in this example), as the error tolerance at this kink, for which the $R(E)$ frontier has its maximum positive curvature. This error tolerance partitions two of three interesting regions of the $R(E)$ frontiers, namely $E > 2.5 \times 10^{-2}$, $E_{\text{cr}} < E < 2.5 \times 10^{-2}$, and $E < E_{\text{cr}}$ (in this example). For $E > 2.5 \times 10^{-2}$, roughness is zero since the spline is allowed enough error that it can be composed of segments that have no roughness.

As E is decreased from 2.5×10^{-2} to E_{cr}, the resulting smoothing splines are required to pass more closely to the given data. In doing so, each successive spline captures more of the roughness of the true function. This is illustrated by splines a1 and a2 in figure 1c and splines n1 and n2 in figure 1d. Note the similarity between the spline fits to the analytic data versus the noisy data; splines a1 and n1 look virtually identical, and splines a2 and n2 look quite similar as well. Thus, when error tolerances are chosen to be larger than the critical error tolerance, E_{cr}, a spline fit to noisy data is quite comparable to a spline fit to the analytic data.

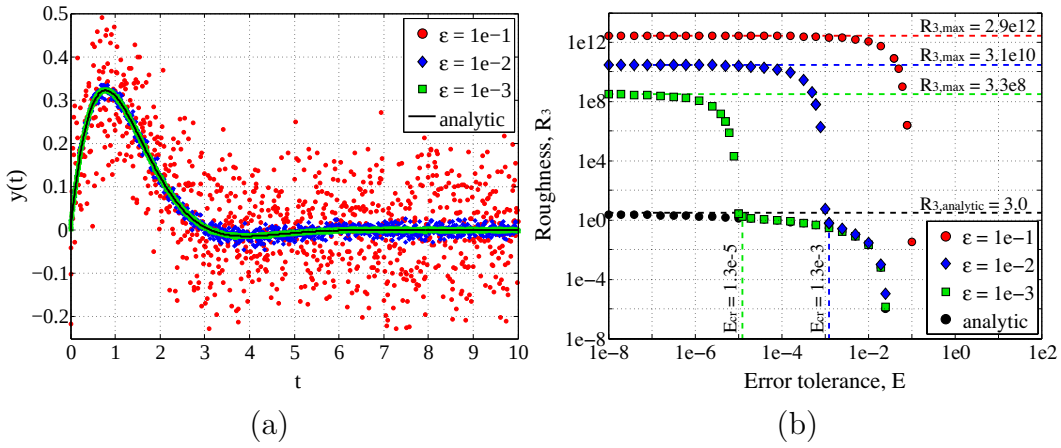

Figure 2: (a) Example analytic function $y(t) = e^{-t}\cdot\sin(t)$ and noisy 'data' $\tilde{y}(t) = y(t)+\mathcal{N}(0,\epsilon^2)$ with $\epsilon = \{10^{-1}, 10^{-2}, 10^{-3}\}$. (b) Roughness of quintic splines, R_3, versus error tolerance, E.

For error tolerances less than the critical value (i.e. $E < E_{\rm cr}$), a smoothing spline fit to noisy data is now required to follow the data so closely that the measurement error is captured by the smoothing spline. In other words, the spline is not permitted enough error tolerance to ignore the measurement error. Consequently, many wiggles are introduced into the spline fit, and the roughness increases by ten orders of magnitude over a relatively small range of E. Splines a3 and n3 (shown in figures 1c and 1d) were computed for an error tolerance just less than that of $E_{\rm cr}$. Note that spline a3 follows the analytic $y(t)$ function more closely than spline a2, whereas spline n3 is quite noisy, because it is attempting to follow the noisy data.

2.1 Roughness and error scaling

We now explore the effect of measurement error on the critical error tolerance and maximum roughness, and we provide scaling arguments that can be used to estimate $E_{\rm cr}$ and $R_{\rm max}$, given the measurement error, ϵ. Conversely, by inspecting the $R(E)$ frontier of an experimental data set, these relations allow us to estimate the measurement error.

Analytical example data with measurement error, $\epsilon = \{10^{-1}, 10^{-2}, 10^{-3}\}$, and their corresponding $R(E)$ frontiers are shown in figures 2a and 2b, respectively. The $\epsilon = 10^{-2}$ data is the same as figure 1. The $\epsilon = 10^{-3}$ data have a lower critical error tolerance than the $\epsilon = 10^{-2}$ data, as shown in figure 2b. The $\epsilon = 10^{-3}$ data more accurately represent the analytic function than the $\epsilon = 10^{-2}$ data, and as a result, the spline fit to the $\epsilon = 10^{-3}$ data at its critical error tolerance more accurately represents the analytic function than the spline fit to the $\epsilon = 10^{-2}$ data at its critical error tolerance. The $R(E)$ frontier corresponding to the $\epsilon = 10^{-1}$ data has no kink, since the noise level is so large that the analytic function cannot be resolved from these data.

To develop scaling arguments for the critical error tolerance and maximum roughness, consider a hypothetical data set, $\tilde{y}(t_i) = (-1)^i \cdot \epsilon$ with $t_i = i \cdot \triangle t$ and $i = 1 \ldots N$, as if the true function were $y(t) = 0$ and this data set represents measurement noise in an average sense.

The critical error tolerance is the minimum error with which the spline still represents the true function (i.e. $s(t) \approx 0$). Thus, the critical error tolerance scales as

$$E_{\rm cr} \sim \sum_{i=1}^{N} |\tilde{y}_i - 0|^2 \triangle t \sim N\epsilon^2 \triangle t \tag{8}$$

In our analytical example, $N = 10^3, \epsilon = 10^{-2}$, and $\triangle t = 10^{-2}$, so by (8), $E_{\rm cr} \sim 10^3 \cdot 10^{-4} \cdot 10^{-2} = 10^{-3}$, which agrees with the computed value of $E_{\rm cr} = 1.3 \times 10^{-3}$ up to an $O(1)$ constant. Note that for the $\epsilon = 10^{-3}$ data, (8) predicts $E_{\rm cr} \sim 10^{-5}$, which also agrees with the computed value of $E_{\rm cr} = 1.3 \times 10^{-5}$ shown in figure 2.

The maximum roughness occurs when the spline passes through every data point. To scale the maximum roughness, we need to scale the second and third derivatives, which we can do using the well-known 'forward divided difference' formulae $\frac{d^2 s(t_i)}{dt^2} = \frac{s_{i+2} - 2s_{i+1} + s_i}{\triangle t^2} \sim \frac{4\epsilon}{\triangle t^2}$ and $\frac{d^3 s(t_i)}{dt^3} = \frac{s_{i+3} - 3s_{i+2} + 3s_{i+1} - s_i}{\triangle t^3} \sim \frac{8\epsilon}{\triangle t^3}$. Thus, the maximum roughness scales by

$$R_{2,\max} = \int_{t_1}^{t_N} \left| \frac{d^2 s}{dt^2} \right|^2 dt \sim N \left(\frac{4\epsilon}{\triangle t^2} \right)^2 \triangle t = 16 N \triangle t^{-3} \epsilon^2 \tag{9}$$

$$R_{3,\max} = \int_{t_1}^{t_N} \left| \frac{d^3 s}{dt^3} \right|^2 dt \sim N \left(\frac{8\epsilon}{\triangle t^3} \right)^2 \triangle t = 64 N \triangle t^{-5} \epsilon^2 \tag{10}$$

For our example $\epsilon = 10^{-2}$ data, (10) predicts $R_{3,\max} \sim 64 \cdot 10^3 \cdot 10^{10} \cdot 10^{-4} = 6.4 \times 10^{10}$, which agrees with the computed value of $R_{3,\max} = 3.1 \times 10^{10}$ up to an $O(1)$ constant.

3 Experimental example

To demonstrate the utility of the present spline fitting method, we use a physical example derived from the high speed video analysis of a sphere falling through water. In this laboratory experiment, a standard billiard ball is dropped into a quiescent pool of water, as shown in figure 3. The velocity and acceleration of the sphere must be determined from the derivatives of the position data. Further details of the physics involved with this water entry problem can be found in references [4, 11].

The goal of the experiment is to compute the force coefficient (i.e. the net hydrodynamic force, normalized by the dynamic pressure force [6])

$$C_F(t) = \frac{F(t)}{\frac{1}{2}\rho [V(t)]^2 A} \tag{11}$$

where $F(t) = ma(t) + mg$ is the net upwards force on the billiard ball, $m = 0.17$ kg is the mass, $a(t) = \frac{d^2 \tilde{y}(t)}{dt^2}$ is the instantaneous acceleration of the sphere, $g = 9.8$ m/s^2 is the acceleration due to gravity, $\rho = 1000$ kg/m^3 is the density of water, $V(t) = \frac{d\tilde{y}(t)}{dt}$ is the instantaneous velocity of the sphere, $A = \pi \left(\frac{d}{2}\right)^2 = 0.0026$ m^2 is the cross-sectional area of the sphere, and $d = 0.057$ m = 2.25 inches is the sphere diameter. In order to compute the force coefficient accurately, we must accurately evaluate the first and second derivatives of the measured $\tilde{y}(t)$ position data.

In the present experiment, a high-speed digital camera acquired $N = 230$ still images at 1000 frame/s ($\triangle t = 0.001$ s) as the sphere plunged into the basin. The position of the center of the billiard ball, $\tilde{y}(t)$, is measured in meters above the quiescent free surface, and time, t, is measured in seconds after impact. (Note that the timing of the camera is accurate to within nanoseconds, so we assume the time of each measurement to be exact.) The image cross-correlation procedure used to acquire the $\tilde{y}(t)$ position data is explained in detail in [4]. Suffice it to say that the procedure yields position with sub-pixel accuracy, and since the optical zoom was 0.762 mm/px, we expect the measurement error to be on the order of $\epsilon \sim O(10^{-1})$ mm.

The experimental $\tilde{y}(t)$ position data are shown in figure 4a. (Note that the abscissa represents *time*, so this is the trajectory of the sphere in time. The sphere falls nearly straight down in space.) The data are very well resolved in time and evolve smoothly; every fifth data point is shown.

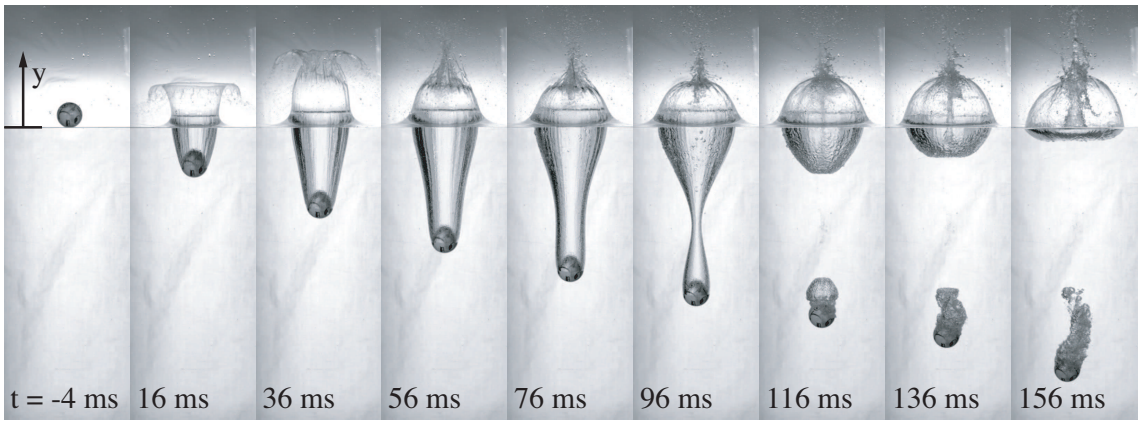

Figure 3: A billiard ball falls into a quiescent pool of water. Position, y, is measured in each timestep, t, by inspection of the images.

3.1 Application of the present spline fitting method

The present spline fitting method is now used to determine the velocity and acceleration from the position data. The minimum roughness versus error tolerance frontier is shown in figure 4b for quintic splines. Quintic splines must be used in order to obtain a smooth second derivative (i.e. acceleration). It should be noted that the choice of spline order implicitly determines the smoothness of the derivatives, so for smooth second derivatives, a quintic spline is necessary, whereas for smooth first derivatives, a cubic spline is sufficient. Figure 4b shows a kink at critical error tolerance $E_{\mathrm{cr}} = 2.5 \times 10^{-9}$. The roughness increases six orders of magnitude as E is decreased below E_{cr}. The maximum roughness, which corresponds to the interpolating spline ($E = 0$), is $R_{3,\max} = 1.7 \times 10^{10}$. It is expected that the smoothing spline corresponding to the critical error tolerance contains very little noise due to measurement error and best approximates the true $y(t)$ curve.

Error tolerance scaling equation (8) can be used to derive an estimate of our experimental measurement error:

$$\epsilon \approx \sqrt{\frac{E_{\mathrm{cr}}}{N \triangle t}} = 0.11,\ \mathrm{mm} \tag{12}$$

which is equivalent to about 0.1% of the sphere diameter. Also note that 0.11 mm = 0.14 px, so this estimate agrees with the assertion that our experimental procedure has sub-pixel accuracy.

Figure 4d shows the force coefficient (11) during the water entry event[2]. For reference, the force coefficient for a sphere of the size and speed in this experiment, when immersed in a free stream of steadily-flowing water, is about 0.2 - 0.5 [6]. The data in figure 4d show that the force coefficient increases from initial water impact until time $t = 83$ ms. Between 83 ms and 113 ms, the force coefficient drops dramatically during the cavity pinch-off process; cavity pinch-off occurs at $t = 98$ ms (just after the sixth image shown in figure 3). A local minimum of force coefficient occurs at $t = 113$ ms, as the lower cavity sheds from the sphere and begins to disintegrate into bubbles. Further discussion of the fluid dynamics can be found in [4]. Using the present spline fitting method was critical in obtaining reliable velocity and acceleration data, which made these force coefficient predictions possible.

[2]Since the sphere slows down during the event, the expected force decreases. Thus, examining the force in physical units is not as insightful as examining the non-dimensional force coefficient.

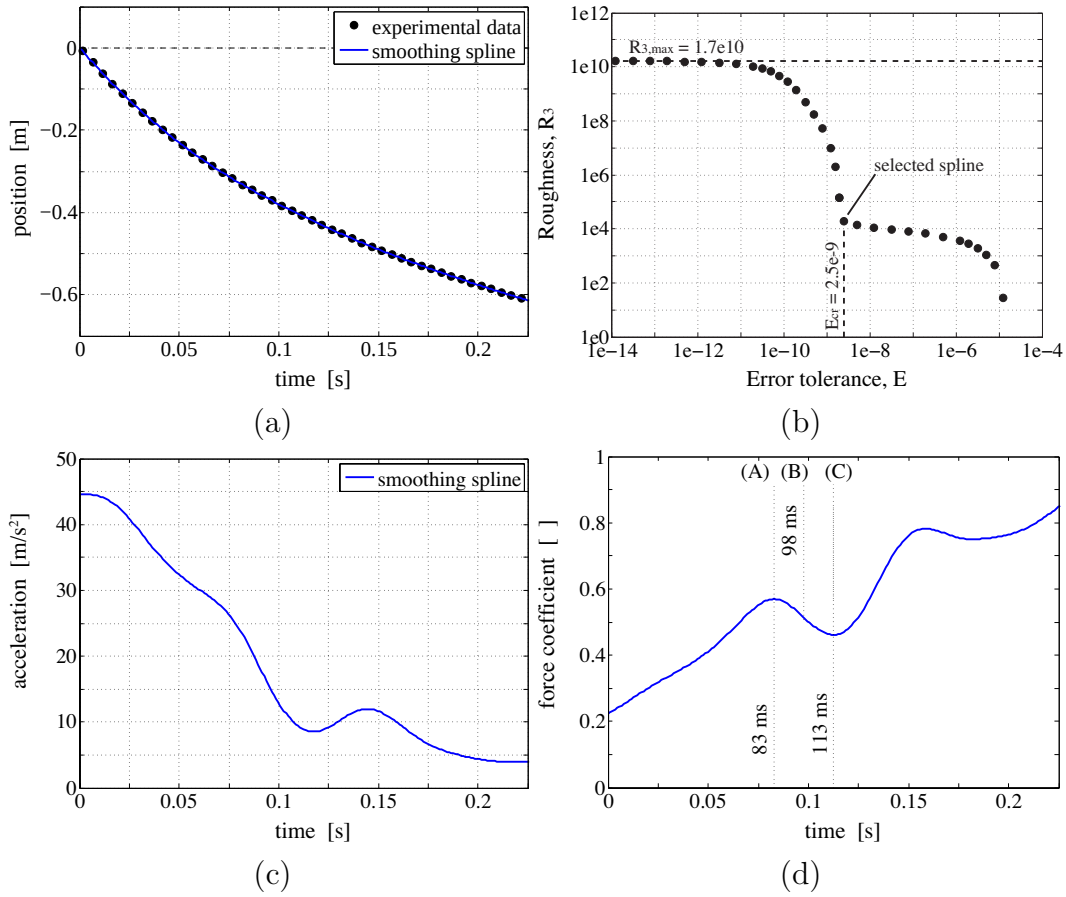

Figure 4: (a) Billiard ball position versus time data, $\tilde{y}(t)$, (every fifth data point shown), and selected smoothing spline fit, $s(t)$. (b) Roughness of quintic smoothing splines, R_3, versus error tolerance, E. The kink in this $R(E)$ frontier is at critical error tolerance, $E_{\text{cr}} = 2.5 \times 10^{-9}$. (c) Second derivative of the smoothing spline, $s''(t)$. (d) Force coefficient, C_F, versus time showing: (A) local maximum force coefficient; (B) pinch-off; and (C) local minimum force coefficient.

3.2 A check for the derivatives $s'(t)$, $s''(t)$, and $s'''(t)$

It is desirable to perform a check on the derivatives of the smoothing spline. For this, we can use the *local least squares* (LLS) regression method [1], which works as follows. To estimate the first derivative at a particular time step, t_i, a line $(a_1 t + a_2)$ is fit to a small window of data surrounding t_i using least squares regression, and the slope of this line (a_1) estimates the derivative at t_i. This process is repeated for windows surrounding each time step, where a larger window yields more smoothing but a less localized estimate of the derivative. Higher order derivatives can also be estimated: A quadratic polynomial fit gives an estimate of the second derivative, a cubic polynomial fit gives an estimate of the third derivative at that time, and so on. Estimates of the first and second derivatives of the billiard ball data are shown in figure 5. These data agree quite well with the derivatives of the smoothing spline, as expected. The local least squares method provides a good estimate of the derivatives of the function, because the general trend of the data surrounding each point is captured by the least squares regression. However, this method does not ensure that the derivative is a smooth function as the window is moved along the data set.

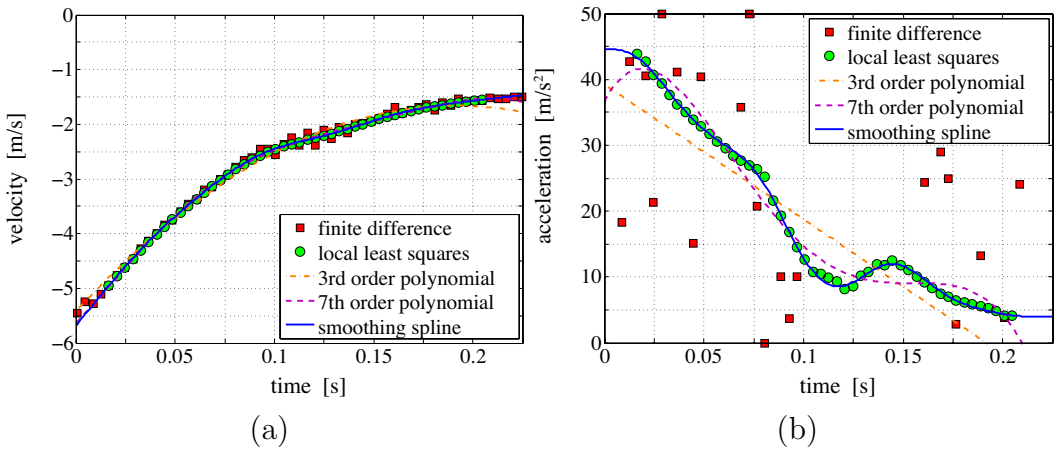

Figure 5: Velocity, $y'(t)$, and acceleration, $y''(t)$, computed by: finite difference, '■'; local least squares, '•'; third-order polynomial least squares fit to the entire data set, '-·-'; seventh-order polynomial least squares fit, '--'; and the smoothing spline with $E = E_{\text{cr}}$, '–'.

Two less accurate methods for estimating the derivatives are also shown in figure 5: least squares regression to the entire data set, and finite differences. The derivatives of a polynomial fit to all the data are inherently questionable, because the fitting parameters depend on the entire data set. Clearly, one cannot assume that the dynamics of our billiard ball during early times (e.g. during cavity formation) are the same as the dynamics during later times (e.g. after cavity collapse). Fitting a single polynomial to all of the data implicitly demands that the physics at all times be the same, which is clearly not true in this experiment, so this method can give misleading results. For example, the second derivatives of 3rd-order and 7th-order polynomials fit to all the data are quite different, and neither agrees with the smoothing spline prediction or local least squares estimate in figure 5b. From the present smoothing spline approach, it is clear that the acceleration of the sphere is not linear throughout its fall.

Finite difference methods amplify measurement noise, yielding poor estimates of derivatives. For example, the central divided difference formula predicts

$$\frac{d\tilde{y}(t_i)}{dt} = \frac{\tilde{y}_{i+1} - \tilde{y}_{i-1}}{2\triangle t} + O(\triangle t^2) = \frac{dy(t_i)}{dt} + O\left(\frac{\epsilon}{\triangle t}\right) + O\left(\triangle t^2\right) \tag{13}$$

where $O(\ \ )$ denotes the order of magnitude of the error in the prediction. For a small timestep, $\triangle t \ll 1$, the measurement error, ϵ, is amplified. The noise is amplified again upon taking each successive derivative, yielding derivatives with unsatisfactorily-large error on the order of

$$\frac{d\tilde{y}}{dt} \sim O\left(\frac{\epsilon}{\triangle t}\right), \frac{d^2\tilde{y}}{dt^2} \sim O\left(\frac{\epsilon}{\triangle t^2}\right), \frac{d^3\tilde{y}}{dt^3} \sim O\left(\frac{\epsilon}{\triangle t^3}\right), \ldots$$

Similarly, all finite difference methods amplify measurement noise, even when a larger time step is used[3]. This error amplification is quite noticeable in the acceleration estimates in figure 5b.

[3]Even if n timesteps are skipped on either side of the data point, the central difference formula predicts

$$\frac{d\tilde{y}(t_i)}{dt} = \frac{y(t_{i+n}) - y(t_{i-n})}{2n\triangle t} + O\left(\frac{\epsilon}{n\triangle t}\right) + O\left(n^2\triangle t^2\right)$$

which may never have satisfactorily-small error.

4 Conclusions

Finding the derivative of noisy data amounts to fitting an analytic curve that best approximates the true function that the data represents. This curve can be differentiated analytically, yielding the required derivatives. We have shown that the derivatives predicted by a best-fit smoothing spline agree well with the local least squares method, which also is a good means to approximate these derivatives. Other methods for finding derivatives, such as finite differences or fitting polynomials to the entire data set yield poor estimates of the desired derivatives.

This paper proposes inspecting the *roughness versus error tolerance efficient frontier* of smoothing splines as a means to select the best spline fit. For high-resolution, high-precision data, this frontier will have a 'kink' at a *critical error tolerance*, E_{cr}, which corresponds to the spline with the minimum error to the data possible without introducing roughness due to the noise in the data.

Two extensions of this work are noteworthy. First, it is well known that the number and location of spline breakpoints strongly affects the fit of a smoothing spline. While these parameters were not considered herein, they certainly do affect the quality of the spline fit. In principle, an efficient frontier exists for each choice of breakpoint sequence, and the overall best spline would be the one picked from the set of splines that are a 'best fit for a given breakpoint sequence'. Another extension of this work is to apply this methodology to fit smoothing spline surfaces to two-dimensional data, such as measurements made along two spatial dimensions or measurements made along one spatial dimension over several time steps, which will allow for smoothing and computing partial derivatives of two-dimensional experimental data.

References

[1] W.S. Cleveland Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Accociation*, 74(367):829–836, 1979.

[2] P. Craven and G. Wahba. Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Num. Mathematik*, 31:377–403, 1979.

[3] Carl de Boor. *A Practical Guide to Splines*. Springer, 1978.

[4] B.P. Epps. *An impulse framework for hydrodynamic force analysis: fish propulsion, water entry of spheres, and marine propellers*. Ph.D. thesis, MIT, February 2010.

[5] Robert Kohn and Craig F. Ansley. A new algorithm for spline smoothing based on smoothing a stochastic process. *SIAM J. Sci. Stat. Comput.*, 8(1):33–48, 1987.

[6] Pijush K. Kundu and Ira M. Cohen. *Fluid Mechanics*. Elsevier Academic Press, third edition, 2004.

[7] S. B. Pope and R. Gadh. Fitting noisy data using cross-validated cubic smoothing splines. *Communications in Statistics - Simulation and Computation*, 17(2):349–376, 1988.

[8] Gerald W. Recktenwald. *Numerical Methods with MATLAB: Implementations and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 2000.

[9] Christian H. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 10:177–183, 1967.

[10] Nicholas A. Teanby. Constrained smoothing of noisy data using splines in tension. *Math Geol*, 39:41–434, 2007.

[11] Tadd Truscott and Alexandra Techet. Water entry of spinning spheres. *Journal of Fluid Mechanics*, 625:135–165, 2009.

[12] Grace Wahba. *Spline Models for Observable Data*. SIAM, 1990.

[13] W. Wecker and C. Ansley. The signal extraction approach to non-linear regression and spline smoothing. *Journal of the American Statistical Association*, 78:81–89, 1983.

[14] S. N. Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *J. R. Statist. Soc. B*, 62(2):413–428, 2000.