Published in final edited form as:

IEEE Access. 2020 ; 8: 196865–196877. doi:10.1109/access.2020.3034077.

Numerical differentiation of noisy data: A unifying multi-objective optimization framework

Floris van Breugel[1], **J. Nathan Kutz**[2], **Bingni W. Brunton**[3]

[1]Department of Mechanical Engineering, University of Nevada, Reno, NV 89557

[2]Department of Applied Math, University of Washington, Seattle, WA, 98195, USA

[3]Department of Biology, University of Washington, Seattle, WA, 98195

Abstract

Computing derivatives of noisy measurement data is ubiquitous in the physical, engineering, and biological sciences, and it is often a critical step in developing dynamic models or designing control. Unfortunately, the mathematical formulation of numerical differentiation is typically ill-posed, and researchers often resort to an *ad hoc* process for choosing one of many computational methods and its parameters. In this work, we take a principled approach and propose a multi-objective optimization framework for choosing parameters that minimize a loss function to balance the faithfulness and smoothness of the derivative estimate. Our framework has three significant advantages. First, the task of selecting multiple parameters is reduced to choosing a single hyper-parameter. Second, where ground-truth data is unknown, we provide a heuristic for selecting this hyper-parameter based on the power spectrum and temporal resolution of the data. Third, the optimal value of the hyper-parameter is consistent across different differentiation methods, thus our approach unifies vastly different numerical differentiation methods and facilitates unbiased comparison of their results. Finally, we provide an extensive open-source Python library pynumdiff to facilitate easy application to diverse datasets (https://github.com/florisvb/PyNumDiff).

Keywords

Numerical differentiation; derivatives; optimization; data-driven modeling

I. Introduction

Derivatives describe many meaningful characteristics of physical and biological systems, including spatial gradients and time rates-of change. However, these critical quantities are often not directly measurable by sensors. Although computing derivatives of analytic equations is straightforward, estimating derivatives from real sensor data remains a significant challenge because sensor data is invariably corrupted by noise [1]. More accurate estimation of derivatives would improve our ability to produce robust diagnostics, formulate accurate forecasts, build dynamic or statistical models, implement control protocols, and

Corresponding author: Floris van Breugel (fvanbreugel@unr.edu).

inform policy making. There exists a large and diverse set of mathematical tools for estimating derivatives of noisy data, most of which are formulated as an ill-posed problem regularized by some appropriate smoothing constraints. However, the level and type of regularization are typically imposed in an *ad hoc* fashion, so that there is currently no consensus "best-method" for producing "best-fit" derivatives.

One particularly impactful application of estimating derivatives is the use of time-series data in modeling complex dynamical systems. These models are of the form $d\mathrm{x} / dt = \dot{\mathrm{x}} = f(\mathrm{x})$, where x is the state of the system. Models of this kind have been integral to much of our understanding across science and engineering [2], including in classical mechanics [3], electromagnetism [4], quantum mechanics [5], chemical kinetics [6], ecology [7] epidemiology [8], and neuroscience [9]-[11]. In some cases, even higher order time derivatives are also crucial for understanding the dynamics [12]. A recent innovation in understanding complex dynamical systems uses data-driven modeling, where the underlying dynamics are learned directly from sensor data using a variety of modern methods [13]-[15]. For this application in particular, a derivative with both small and unbiased errors is crucial for learning interpretable dynamics.

In principle, the discrete derivative of position can be estimated as the finite difference between adjacent measurements. If we write the vector of all noiseless positions in time measured with timestep Δt as x, then

$$\dot{\mathrm{x}}_k = \frac{\mathrm{x}_{k+1} - \mathrm{x}_k}{\Delta t}, \qquad (1)$$

where k indexes snapshots in time. In reality, however, only noisy measurements **y** are available,

$$\mathrm{y} = \mathrm{x} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ represents measurement noise. Here we will assume $\boldsymbol{\eta}$ is zero-mean Gaussian noise with unknown variance. Even with noise of moderate amplitude, a naïve application of Eq. (1) produces derivative estimates that are far too noisy to be useful (Fig. 1A). Thus, more sophisticated methods for data smoothing and/or differentiation of noisy time series measurements of position **y** are required.

Although smoothing mitigates the errors, it can also introduce biases. Our goal in this paper is to develop a general approach for methodically choosing parameters that balance the need to minimize both error and bias. We use $\hat{\mathrm{x}}$ and $\dot{\hat{\mathrm{x}}}$ to denote the smoothed *estimates* of the position and its derivative computed from **y**, respectively. To evaluate the quality of these estimates, we compare these estimates to the *true* discrete time position and its derivative, x and $\dot{\mathrm{x}}$. Developing approaches for estimating $\dot{\hat{\mathrm{x}}}$ from noisy measurements **y** has been the focus of intense research for many decades. Despite the diversity of methods that have been developed, only a few studies have performed a comprehensive comparison of their performance on different types of problems [1], [16], [17].

In this paper, we tackle the challenge of parameter selection by developing a novel, multi-objective optimization framework for choosing parameters to estimate the derivative of noisy data that balances two independent metrics. Our approach minimizes a loss function consisting of a weighted sum of two metrics computed from the derivative estimate: the faithfulness of the integral of the derivative and its smoothness. We suggest these metrics as proxies for minimizing the error and bias of the estimated derivative, and we show that sweeping through values of a single hyper-parameter γ produces derivative estimates that generally trace the Pareto front of solutions that minimize error and bias. Importantly, this optimization framework assumes no knowledge of the underlying true derivative and reduces the task of selecting many parameters of any differentiation algorithm to solving a loss function with a single hyper-parameter. Furthermore, we show that the value of the hyper-parameter is nearly universal across four different differentiation methods, making it possible to compare the results in a fair and unbiased way. For real-world applications, we provide a simple heuristic to determine a value of γ that is derived from the power spectrum and temporal resolution of the data. All of the functionality described in this paper is implemented in an open-source Python toolkit pynumdiff, which is found here: https://github.com/florisvb/PyNumDiff.

II. Motivation for error metrics

What is a “good” estimate of a derivative? Let us start by considering a toy system with synthetic measurement noise, where we are able to evaluate the quality of an estimated derivative by comparing to the true, known derivative. We consider two metrics for evaluating the quality of a derivative Fig. 1B-D; later, we use these same metrics to evaluate the performance of our optimization framework, which does not have access to the ground truth.

First, the most intuitive metric is how faithfully the estimated derivative $\hat{\dot{x}}$ approximates the actual derivative $\dot{x}$. We can measure this using the root-mean-squared error,

$$\mathrm{RMSE}(\hat{\dot{x}}, \dot{x}) = \|(\hat{\dot{x}} - \dot{x})\|_2, \tag{2}$$

where $\|\cdot\|_2$ is the vector 2-norm. If the data are very noisy, a small RMSE can only be achieved by applying significant smoothing. However, smoothing the data often attenuates sharp peaks in the data and results in underestimating the magnitude of the derivative.

To measure the degree to which the derivative estimate is biased due to underestimates of the actual derivative, we calculate the square of the Pearson’s correlation coefficient, R^2, between the errors $(\hat{\dot{x}} - \dot{x})$ and the actual derivative $\dot{x}$. We refer to this metric as the *error correlation*, which is bounded between 0 and 1. Small error correlations imply that the imposed dynamics of the differentiation method (e.g. filtering) minimally influenced the derivative estimate; therefore, the method of estimating derivatives would have minimal impact on any models that are constructed using these estimates. Conversely, large error correlations imply that the estimate is significantly influenced by the dynamics of the differentiation method and typically correspond to very smooth estimates. In the limit where the derivative estimate is a horizontal line, the error correlation takes on a value of unity.

Other metrics that measure the smoothness, for example the total variation or tortuosity, may be substituted for error correlation [1]; however, these metrics are harder to interpret. For instance, if the true derivative is very smooth, a low total variation is desired, whereas if the true derivative is quite variable, a high total variation would correspond to an accurate derivative. In contrast, a low error correlation is desirable for any true derivative.

For many datasets, the RMSE and error correlation metrics define a Pareto front, where no single parameter choice minimizes both values (Fig. 1B). Furthermore, the minimal RMSE can be achieved with a variety of different error correlations. The most suitable parameter set depends on the application of the estimated derivative: is a non-smooth derivative with minimal bias preferred (Fig. 1B-D: teal), or one that is smooth, but biased (Fig. 1B-D: brown). We suggest that, for most purposes, the estimated derivative that balances these metrics (Fig. 1B-D: blue and red) serves as a reasonable starting point.

III. Methods for Numerical Differentiation

A large variety of methods for numerical differentiation exist, and a complete review of them all is beyond the scope of this paper. Instead, we have selected four differentiation methods (Table I), which make different assumptions and represent different approaches to computing the derivative including both global and local methods [1], to showcase the universal application of our optimization framework.

One common approach to manage noisy data is to apply a smoothing filter to the data itself, followed by a finite difference calculation. In this family of differentiation methods, we chose to highlight the **Butterworth filter** [18], which is a global spectral method with two parameters: filter order and frequency cutoff.

The second family of methods relies instead on building a local model of the data through linear regression. A common and effective approach involves making a sliding polynomial fit of the data [19], often referred to as locally estimated scatterplot smoothing (LOESS) [20]. An efficient approach for accomplishing the same calculations is the **Savitzky-Golay filter**, which builds the polynomial model in the frequency domain [21], [22]. The Savitzky-Golay filter has two parameters: window size and polynomial order. By default, a Savitzky-Golay filter provides a jagged derivative because the polynomial models can change from one window to the next, so here we also apply some smoothing by convolving the result with a Gaussian kernel. This smoothing adds a third parameter: a smoothing window size.

The third family we consider is the **Kalman filter** [23]-[25]. The Kalman filter is most effective when models of the system and of the noise characteristics are known. Our focus here is the case where neither is known, so we chose to highlight a constant acceleration forward-backward Kalman smoother [26] with two parameters: the model and noise covariances.

Finally, we consider an optimization approach to computing derivatives with the **total variation regularization** (TVR) method [27], [28]. One advantage of the TVR methods is that there is only a single parameter, which corresponds to the smoothness of the derivative estimate. TVR derivatives are not as widely used as the other three methods we highlight, so we provide a brief overview here. Solving for the TVR derivative involves first finding $\hat{x}$ and its corresponding finite-difference derivative $\hat{\dot{x}}$ (calculated according to Eq. 1) that minimize the following loss function,

$$L = \|y - \hat{x}\|_2 + \gamma * TV(\hat{\dot{x}}) . \tag{3}$$

Here TV is the total variation,

$$TV(\hat{\dot{x}}) = \frac{1}{m}\left\|\hat{\dot{x}}_{0:m-1} - \hat{\dot{x}}_{1:m}\right\|_1, \tag{4}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm and m is the number of time snapshots in the data. The single parameter for this method is γ, and larger values result in smoother derivatives. If γ is zero, this formulation reduces to a finite difference derivative.

Solutions for TVR $\hat{\dot{x}}$ can be found with an iterative solver [28]. Because both components of the loss function Eq. (3) are convex, we can also solve for $\hat{\dot{x}}$ using convex optimization tools, such as cvxpy [29], and with a convex solver, such as MOSEK [30]. The two methods are equivalent, if the iterative solver is repeated sufficiently many times.

The convex solution to penalizing the first order difference in time, as in Eq. (4), results in a piece-wise constant derivative estimate. By offloading the calculations to a convex optimization solver, however, we can easily penalize higher order derivatives by replacing the 1^{st} order finite difference derivative $\hat{\dot{x}}$ in Eq. (3) with a 2^{nd} order $(\hat{\ddot{x}})$ or 3^{rd} order $(\hat{\dddot{x}})$ finite difference derivative. Penalizing higher-order time derivatives results in smoother derivative estimates. For example, penalizing the 2^{nd} order derivative results in a piece-wise linear derivative estimate, whereas penalizing the 3^{rd} order derivative, also known as the *jerk*, results in a smooth estimate. In this paper, we will use the total variation regularized on the jerk (TVRJ). For large datasets, solving for the TVRJ derivative is both computationally expensive and can accumulate small errors. To manage the size of the optimization problem, it is possible to solve for the TVRJ derivative in sliding windows. In practice, we found that using window sizes of 1000 sufficiently reduces the error accumulation, and using a stride of 200 ensures smooth transitions from one window to the next.

IV. Computing derivatives of noisy data with no ground truth

With noisy data collected in the real world, no ground truth is accessible. The RMSE and error correlation metrics described in the previous section cannot be calculated and used to optimize parameter choices, so the parameter selection is an ill-posed problem. Even so—somehow—parameters must be chosen. In this section, we propose a general approach for choosing parameters and show that for a wide range of problems, noise levels, time resolutions, and methods, our approach yields reasonable derivative estimates without the need for hyper-parameter tuning.

A. Optimization framework without ground truth derivatives

Given noisy position measurements y, we seek to estimate the derivative in time of the dynamical system that underlies the measurements $\hat{\dot{x}}$. When the ground truth $\dot{x}$ is unknown, we propose choosing the set of parameters Φ (for any given numerical algorithm, including those enumerated in Table I) that minimize the following loss function, which is inspired by Eq. (3),

$$L = \mathrm{RMSE}\left(\mathrm{trapz}(\hat{\dot{x}}(\Phi)) + \mu, \mathrm{y}\right) + \gamma\left(TV(\hat{\dot{x}}(\Phi))\right), \tag{5}$$

where trapz(·) is the discrete-time trapezoidal numerical integral, μ resolves the unknown integration constant,

$$\mu = \frac{1}{m}\sum_{k=0}^{m}\left(\mathrm{trapz}(\hat{\dot{x}}(\Phi)) - \mathrm{y}\right), \tag{6}$$

and γ is a hyper-parameter. Note that this formulation has a single hyper-parameter γ, and a heuristic for choosing γ is introduced in the following section.

The first term of the loss function in Eq. (5) promotes faithfulness of the derivative estimate by ensuring that the integral of the derivative estimate remains similar to the data, whereas the second term encourages smoothness of the derivative estimate. If γ is zero, the loss function simply returns the finite difference derivative. Larger values of γ will result in a smoother derivative estimate.

This loss function effectively reduces the set of parameters Φ (which ranges between 1 and 3 or more, depending on the method) to a single hyper-parameter γ. Unfortunately, L is not convex, but tractable optimization routines can be used to solve for the set of Φ that minimize L. Here we use the Nelder-Mead method [31], a downhill simplex direct search method that works well for nonlinear optimization problems, as implemented in SciPy [32]. To prevent the optimization from converging on incorrect minima, we used with multiple initial conditions.

B. Heuristics for automated hyper-parameter tuning of γ

The advantages of our loss function in Eq. (5) are that it does not require any ground truth data, and it simplifies the process of choosing parameters by reducing all the parameters associated with any given method for differentiation to a single hyper-parameter γ corresponding to the how smooth the resulting derivative should be. To understand the qualities of the derivative estimates resulting from parameters selected by our loss function, we begin by analyzing the derivative estimates of noisy sinusoidal curves using the Savitzky-Golay filter and return to our original metrics, RMSE and error correlation to evaluate the results.

Interestingly, sweeping through values of γ results in derivative estimates with RMSE and error correlation values that generally follow the Pareto front defined by all possible derivative estimates for that given method (Fig. 2A). Which of these derivative estimates is best depends on the intended use of the derivative; nevertheless, we suggest that a good general purpose derivative is one that corresponds with the elbow in the lower left corner of the stereotypical curve traced by a sweep of γ in the RMSE vs. error correlation space (the star-shaped markers in Fig. 2A). This point often, but not always, corresponds to the lowest RMSE (for example, see Fig. 1). Although in many cases a quantitatively better derivative estimate than the one found by our loss function does exist (the gray dots in Fig. 2A that lie left of the star), the qualitative differences between these two derivative estimates are generally small (Fig. 2A middle row).

In practice, the need to choose even a single parameter can be time consuming and arbitrary. To alleviate these issues, we derive an empirical heuristic to guide the choice of γ that corresponds with the elbow of the Pareto front. We found that the best choice of γ is dependent on the frequency content of the data. To characterize this relationship, we evaluated the performance of derivative estimates achieved by a Savitzky-Golay filter by sweeping through different values of γ for a suite of sinusoidal data with various frequencies (f), noise levels (additive white (zero-mean) Gaussian noise with variance σ^2), temporal resolutions (Δt), and dataset lengths (in time steps, L) (Fig. 2A-B).

To describe this empirical relationship between the optimal choice of γ and quantitative features of the data, we first considered an all-inclusive multivariate log-linear model,

$$\log(\gamma) = \alpha_1 \log(f) + \alpha_2 \log(\Delta t) + \alpha_3 \log(\sigma) + \alpha_4 \log(L) + \alpha_5 . \tag{7}$$

Fitting the data (Fig. 2B triangles) to this model with ordinary least squares resulted in an R^2 = 0.76, suggesting that, in many cases, it is feasible to use this rule to determine a reasonable guess for γ. To ensure that our decision to take the logarithm of each input and output was appropriate, we tried all possible combinations linear and log-transformed inputs and outputs. Taking the logarithm of each input and output resulted in the highest adjusted R^2, indicating that it explains the largest percentage of variance. Table II provides the coefficients (α_k) and associated p-values for each of the four terms and intercept. From this analysis we can conclude that the magnitude of measurement noise in the data is *not* an important predictor of γ. We note, however, that here we have assumed that the magnitude of noise does not change within a time-series dataset.

Eliminating the unnecessary terms from our model results in slightly adjusted coefficients, provided in Table III. In short, the optimal choice of γ, assuming that both low RMSE and low error correlation are valued, can be found according to the following relationship:

$$\log(\gamma) = -1.6 \log(f) - 0.71 \log(dt) - 5.1 . \tag{8}$$

We analyze the performance of our loss function and heuristic with respect to a broad suite of representative synthetic problems. Real world data takes on a much greater diversity of shapes than the sinusoidal timeseries we used to derive the heuristic for choosing γ given in Eq. (8). Because it is difficult to define a clear quantitative description of the range of shapes that real data might take on (such as frequency for a sinusoidal function), we first examine differentiating one component of a non-periodic Lorenz system [33] (Fig. 3) with four different levels of added noise and temporal resolutions. From the power spectra, we select a frequency corresponding to the frequency where the power begins to decrease and the noise of the spectra increases. Although somewhat arbitrary, this approach (in conjunction with Eq. (8)) allows us to use a standard signal processing tool to quickly determine a choice of γ. Our method produces reliable derivatives without further tuning in each case except high noise and low temporal resolution (Fig. 3, fourth row), which is not surprising considering the low quality of the data.

Next we consider four other synthetic problems, all with similarly effective results (Fig. 3). For the logistic growth problem, the curve traced by our loss function takes on a more complicated shape, perhaps because the characteristics of data vary substantially across time. Still, our heuristic results in a good choice of parameters that correspond to an accurate derivative. For the triangle wave, the loss function does a good job of tracing the Pareto front, and the heuristic selects an appropriate value of γ, yet the resulting derivative does show significant errors. This is likely due to two reasons. First, the Savitzky-Golay filter is designed to produce a smooth derivative, rather than a piece-wise constant one. Second, the frequency content of the data varies between two extremes, near-zero, and near-infinity. For the sum of sines problem, selecting the appropriate frequency cutoff is more straightforward than the previous problems, as we can simply choose a frequency shortly after the high frequency spike in the spectra. The final problem is a time-series resulting from a simulated dynamical system controlled by a proportional-integral controller subject to periodic disturbances. This data is a challenging problem for numerical differentiation, as the position data almost appears to be a straight line but does contain small variations. Our loss function does an excellent job of tracing the Pareto front in this case, and our heuristic results in an appropriate choice of γ.

C. Direct comparison of differentiation methods

We examine how our loss function and heuristic for choosing γ might perform on other differentiation methods beyond the Savitzky-Golay filter. Figure 4 shows that for a noisy Lorenz system, the possible solution space is similar for all four methods we highlighted earlier, and our loss function achieves a similar Pareto front in each case. Note that although the Savitzky-Golay and Butterworth filters both operate in the frequency domain, the Kalman smoother and TVRJ methods do not.

Interestingly, for all four differentiation methods, the possible solutions (the gray dots), and in particular their Pareto front, are quite similar, with the exception of the TVRJ method. This deviation may be because the TVRJ method only contains a single parameter. Our loss function, which defines the colored curves in the RMSE vs error correlation space, results in similar curves for each method, each of which follows the Pareto front quite closely. Although there are some differences in the location along the Pareto front that our heuristic selects as the optimal choice for each method, the resulting derivative estimates are qualitatively quite similar. A close comparison of the curves defined by the loss function, and the points selected by the heuristic, suggest that the Kalman and TVRJ methods produce slightly more accurate derivative estimates with a lower error correlation. However, looking at the resulting derivatives we see that the regions where the derivative estimates have high errors, all four estimates exhibit similar errors, suggesting that these errors may be a result of the data, not the method.

These results suggest that our optimization framework is universal across different methods, a claim further supported by its performance across a range of synthetic problems (Fig. 5. The most significant result of this analysis is that all four methods, despite being very different in their underlying mathematics, behave similarly under both our loss function and heuristic for choosing γ across a wide range of data. Even in the case where they disagree on a quantitative level (second row, low temporal resolution Lorenz data), and the Savitzky-Golay filter appears to provide the estimate with the lowest error correlation, the resulting derivative estimates are in fact qualitatively quite similar.

Taking a closer look at the errors in the derivative estimates across the range of toy problems shown in Fig. 5 reveals a subtle point about the limitations of the differentiation methods we highlight here. For all four methods, the errors in the derivative estimates are largest for the triangle problem, and to a lesser extent the proportional-integral control problem. These errors likely stem from two particular challenges. First, the frequency content of the data is very heterogeneous: it is near zero between the peaks and valleys, and near infinite at the peaks and valleys. Furthermore, the frequency of the oscillations for the triangle increase with time. Second, all four of the methods we highlighted here are designed to provide smooth derivatives, whereas the true derivative for the triangle problem is piece-wise constant. If this were known from the outset, it might be more effective to choose a method that is designed to return piece-wise constant derivatives, such as the total variation regularized on the 1^{st} derivative.

V. Demonstrations on real-world data

The real value of our multi-objective optimization framework is its straightforward application to real, noisy data where no ground truth data is available. Here we provide two such examples: differentiation of the new confirmed daily cases in the United States of COVID-19, the disease caused by SARS-CoV-2 (Fig. 6), and differentiation of gyroscope data from a downhill ski (Fig. 7). In both examples, we examine the power spectra of the data to choose a cutoff frequency that corresponds to the start of the dropoff in power. This cutoff frequency, in conjunction with the time resolution of the data, are then used as inputs to our heuristic described by Eq. (8) to determine an optimal value of γ. With γ chosen, we minimize our loss function from Eq. (5) to find the optimal parameters for numerical differentiation.

The year 2020 has seen a dramatic growth of the prevalence of a novel coronavirus, SARS-CoV-2, which causes the disease known as COVID-19. Estimating and understanding the rate of increase of disease incidence is important for guiding appropriate epidemiological, health, and economic policies. In the raw data ([34], https://github.com/CSSEGISandData/COVID-19) for the raw new confirmed daily cases of COVID-19, Fig. 6A) there is a clear oscillation with a period of one week, most likely due to interruptions in testing and reporting during weekends. As such, we selected a lower cutoff frequency of 2 months, corresponding to the beginning of the steep drop off in the power spectra (Fig. 6B). If the weekly oscillations were important, one could just as easily select a cutoff frequency of 1/week. Our heuristic for choosing γ was based on sinusoidal data with a limited domain of time resolutions ranging from 0.001 to 0.1 seconds, so we scaled the time step units of the COVID-19 data to be close to this range, using $dt = 1$ day, rather than 86,400 seconds. Our chosen cutoff frequency yielded a value of $\gamma = 4.1$.

Using this same value of γ for each of the four differentiation methods under consideration resulted in very similar smoothed daily case estimates and derivatives, except during the final 2 weeks (Fig. 6C). In the final week, the raw data shows a fast decrease in the number of daily cases. Three out of the four methods follow this quick dive, whereas the Savitzky-Golay filter is less quick to respond. This is likely because the Savitzky-Golay filter operates on sliding windows. At the end of the data stream these sliding windows are not symmetric about the point being estimated and will therefore weight the data to the left of the end point more. Such edge artifacts are common among smoothing methods, and each method will have its own challenges. One potential solution would be to include specific boundary conditions, as has been done recently for spline fits [35]. A more subtle difference between the methods is that the Butterworth filter appears to preserve a larger remnant of the weekly oscillations seen in the raw data, likely due to either a too high a filter order, or too high of a cutoff frequency. The differences in estimates highlights an important application of our method, which facilitates easy and fair comparison between different smoothing methods. Where these methods disagree, it is clear that none of the estimates can be trusted.

Finally, we consider angular velocity data collected from a gyroscope attached to a downhill ski over one minute of descent (Fig. 7A) (ICM-20948, SparkFun; Wildcat Ski, Moment Skis). This type of data is representative of kinematic data that might be collected during experiments with robots or animals, which might be used to construct data-driven models of their dynamics [36]. From the power spectrum, we chose a cutoff frequency of 0.2 Hz (Fig. 7B). This selection together with the time resolution of 0.0009 seconds yielded an optimal value of $\gamma = 11.5$ using our heuristic. We calculated the smoothed angular velocity and acceleration estimates using a Savitzky-Golay filter (Fig. 7D-F). The other methods showed similar results (not shown for visual clarity), though the total variation method is not recommended for large datasets like this one due to the compounding computational costs.

VI. Conclusion

In summary, this paper develops a principled multi-objective optimization framework to provide clear guidance for solving the ill-posed problem of numerical differentiation of noisy data, with a particular focus on parameter selection. We define two independent metrics for quantifying the quality of a numerical derivative estimate of noisy data: the RMSE and error correlation. Unfortunately, neither metric can be evaluated without access to ground truth data. Instead, we show that the total variation of the derivative estimate, and the RMSE of its integral, serve as effective proxies, as the solutions resulting from these metrics generally trace the pareto front of the solutions resulting from the actual RMSE and error correlation (e.g. compare the violet trace and gray dots in Fig. 3). We then introduced a novel loss function that balances these two proxies, reducing the number of parameters that must be chosen for any given numerical differentiation method to a single universal hyperparameter, which we call γ. Importantly, the derivative estimates resulting from a sweep of γ lie close to the Pareto front of all possible solutions with respect to the true metrics of interest. Although different applications may require different values of γ to produce more smooth or less biased derivative estimates, we derive an empirical heuristic for determining a general purpose starting point for γ given two features that can easily be determined from timeseries data: the cutoff frequency and time step. Our method also makes it possible to objectively compare the outputs for different methods. We found that for each problem that we tried, the four differentiation methods we explored in depth, including both local and global methods, all produce qualitatively similar results.

In our loss function we chose to use the RMSE of the integral of the derivative estimate and the total variation of the derivative estimate as our metrics. However, our loss function can be extended to a more general form,

$$L = M_1(\hat{\dot{x}}, \dot{x}) + \gamma_2 M_2(\hat{\dot{x}}, \dot{x}) + \cdots + \gamma_p M_p(\hat{\dot{x}}, \dot{x}), \tag{9}$$

where M_1, M_2, $\cdots$, M_p represent p different metrics that could be used, balanced by $p-1$ hyper-parameters. Alternative metrics include, for example, the tortuosity of the derivative estimate, the error correlation between the data and the integral of the derivative estimate, a metric describing the distribution of the error between the data and the integral of the derivative estimate. Depending on the qualities of the data and the specific application, different sets of metrics may be suitable as terms in the loss function.

Our loss function makes three important assumptions that future work may aim to relax. The first is that we assume the data has consistent zero-mean Gaussian measurement noise. How sensitive the loss function and heuristic are to outliers and other noise distributions remains an open question. It is possible that once we include other noise models, we will find differences in the behavior of differentiation methods. The second major limitation is that our loss function finds a single set of parameters for a given time series. For data where the frequency content dramatically shifts over time, it may be better to use time-varying parameters. Presently, this is limited by our current implementation, which relies on a computationally expensive optimization step. Future efforts may focus on ways to improve the efficiency of these calculations. Finally, we have focused on single dimensional time-series data. In principle, our proposed loss function can be used with multi-dimensional data, such as 2- and 3-dimensional spatial data, with only minor modifications.

By simplifying the process of parameter selection for numerical differentiation to the selection of a single hyper-parameter, our approach makes it feasible to directly compare the performance of different methods within a given application. One particular application of interest is that of data-driven model discovery. Methods such as sparse identification of nonlinear dynamics (SINDy) [14], for example, rely directly on numerical derivative estimates, and the characteristics of these estimates can have an important impact on the resulting models. Using our method, it is now tractable to systematically investigate the collection of data-driven models learned from estimated derivatives of different smoothness and explore their impact on the models.

Acknowledgements

We are grateful for helpful discussions with Steve Brunton and Pierre Karashchuk. FvB acknowledges support from a Moore/Sloan Data Science and Washington Research Foundation Innovation in Data Science Postdoctoral Fellowship, a Sackler Scholarship in Biophysics, and the National Institute of General Medical Sciences of the National Institutes of Health (P20GM103650). JNK acknowledges support from the Air Force Office of Scientific Research (FA9550-19-1-0011). BWB acknowledges support from the Washington Research Foundation, the Air Force Office of Scientific Research (FA9550-19-1-0386), and the National Institute of Health (1R01MH117777).

Biography

Floris van Breugel received the B.S. degree in biological engineering from Cornell University, Ithaca, NY, in 2008, and the Ph.D. degree in control and dynamical systems from the California Institute of Technology, Pasadena, CA, in 2014. He is currently an assistant Professor of mechanical engineering at the University of Nevada, Reno.

J. Nathan Kutz received the B.S. degrees in physics and mathematics from the University of Washington, Seattle, WA, in 1990, and the Ph.D. degree in applied mathematics from Northwestern University, Evanston, IL, in 1994. He is currently a Professor of applied mathematics, adjunct professor of physics, mechanical engineering, and electrical engineering, and a senior data science fellow with the eScience institute at the University of Washington.

Bingni W. Brunton received the B.S. degree in biology from the California Institute of Technology, Pasadena, CA in 2006 and the Ph.D. degree in molecular biology and neuroscience from Princeton University, NJ in 2012. She is currently an associate professor of biology, adjunct associate professors of computer science and engineering and applied mathematics, and a data science fellow of the eScience Institute.

References

[1]. Ahnert K and Abel M, “Numerical differentiation of experimental data: local versus global methods,” Computer Physics Communications, vol. 177, pp. 764–774, 11 2007.

[2]. Lin C-C and Segel LA, Mathematics applied to deterministic problems in the natural sciences. SIAM, 1988.

[3]. Goldstein H, Poole C, and Safko J, Classical mechanics. Addison-Wesley, 2002.

[4]. Jackson JD, Classical electrodynamics. John Wiley & Sons, 2007.

[5]. Griffiths DJ and Schroeter DF, Introduction to quantum mechanics. Cambridge University Press, 2018.

[6]. Masel RI, Chemical Kinetics and Catalysis. John Wiley & Sons, 2001.

[7]. Kot M, Elements of mathematical ecology. Cambridge University Press, 2001.

[8]. Rothman KJ, Greenland S, and Lash TL, Modern epidemiology. Lippincott Williams & Wilkins, 2008.

[9]. Roth E, Sponberg S, and Cowan N, "A comparative approach to closed-loop computation," Current Opinion in Neurobiology, vol. 25, pp. 54–62, 4 2014. [PubMed: 24709601]

[10]. Madhav MS and Cowan NJ, "The synergy between neuroscience and control theory: The nervous system as inspiration for hard control challenges," Annual Review of Control, Robotics, and Autonomous Systems, vol. 3, pp. 243–267, 5 2020.

[11]. Boker S, Deboeck P, Schiller C, and Keel P, "Generalized local linear approximation of derivatives from time series," Statistical methods for modeling human dynamics : An interdisciplinary dialogue, 161-178 (2010), 1 2010.

[12]. Lin DC, McGowan CP, Blum KP, and Ting LH, "Yank: the time derivative of force is an important biomechanical variable in sensorimotor systems," The Journal of Experimental Biology, vol. 222, p. jeb180414, 9 2019. [PubMed: 31515280]

[13]. Schmidt M and Lipson H, "Distilling free-form natural laws from experimental data," Science, vol. 324, pp. 81–85, 4 2009. [PubMed: 19342586]

[14]. Brunton SL, Proctor JL, and Kutz JN, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proceedings of the National Academy of Sciences, vol. 113, no. 15, pp. 3932–3937, 2016.

[15]. Daniels BC and Nemenman I, "Automated adaptive inference of phenomenological dynamical models," Nature Communications, vol. 6, 8 2015.

[16]. Walker JA, "Estimating velocities and accelerations of animal locomotion: a simulation experiment comparing numerical differentiation algorithms," The Journal of Experimental Biology, vol. 201, pp. 981–995, 1998.

[17]. Crenshaw MMHC, Ciampaglio CN, "Analysis of the three-dimensional trajectories of organisms: estimates of velocity, curvature and torsion from positional information," The Journal of Experimental Biology, vol. 203, pp. 961–982, 1998.

[18]. Butterworth S, "On the theory of filter amplifiers," Experimental Wireless and the Wireless Engineer, vol. 7, pp. 536–541, 1930.

[19]. Belytschko T, Krongauz Y, Organ D, Fleming M, and Krysl P, "Meshless methods: An overview and recent developments," Computer Methods in Applied Mechanics and Engineering, vol. 139, pp. 3–47, 12 1996.

[20]. Harrell F, Regression modeling strategies : with applications to linear models, logistic and ordinal regression, and survival analysis. Cham: Springer, 2015.

[21]. Schafer R, "What is a savitzky-golay filter? [lecture notes]," IEEE Signal Processing Magazine, vol. 28, pp. 111–117, 7 2011.

[22]. Savitzky A and Golay MJE, "Soothing and differentiation of data by simplified least squares procedures," Anal. Chem, vol. 36, pp. 1627–1639, 1964.

[23]. Kalman RE, "A new approach to linear filtering and prediction problems," Journal of Basic Engineering, vol. 82, pp. 35–45, 3 1960.

[24]. Zarchan P, Fundamentals of Kalman filtering : a practical approach. Reston, VA: American Institute of Aeronautics and Astronautics, Inc, 2015.

[25]. Aravkin A, Burke JV, Ljung L, Lozano A, and Pillonetto G, "Generalized kalman smoothing: Modeling and algorithms," Automatica, vol. 86, pp. 63–86, 12 2017.

[26]. Crassidis JL and Junkins JL, Optimal Estimation of Dynamic Systems, Second Edition (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science). Chapman & Hall/CRC, 2nd ed., 2011.

[27]. E. F. Rudin Leonid I., Osher Stanley, “Nonlinear total variation based noise removal algorithms,” Physica D: Nonlinear Phenomena, vol. 60, pp. 259–268, 1992.

[28]. Chartrand R, “Numerical differentiation of noisy, nonsmooth data,” ISRN Applied Mathematics, p. 164564, 2011.

[29]. Diamond S and Boyd S, “CVXPY: A Python-embedded modeling language for convex optimization,” Journal of Machine Learning Research, vol. 17, no. 83, pp. 1–5, 2016.

[30]. ApS M, The MOSEK optimization API for Python. Version 8.1, 2018.

[31]. Nelder JA and Mead R, “A simplex method for function minimization,” The Computer Journal, vol. 7, pp. 308–313, 1 1965.

[32]. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Jarrod Millman K, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey C, Polat I, Feng Y, Moore DW, Vand erPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, and S. . . Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” Nature Methods, 2020.

[33]. Lorenz EN, “Deterministic nonperiodic flow,” Journal of the Atmospheric Sciences, vol. 20, pp. 130–141, 3 1963.

[34]. Dong E, Du H, and Gardner L, “An interactive web-based dashboard to track COVID-19 in real time,” The Lancet Infectious Diseases, vol. 20, pp. 533–534, 5 2020. [PubMed: 32087114]

[35]. Tong M, Zhang H, Ott D, Chu W, and Song J, “Applications of the spline filter for areal filtration,” Measurement Science and Technology, vol. 26, p. 127002, 11 2015.

[36]. Karashchuk P, Rupp KL, Dickinson ES, Sanders E, Azim E, Brunton BW, and Tuthill JC, “Anipose: a toolkit for robust markerless 3d pose estimation,” bioRxiv, 2020.

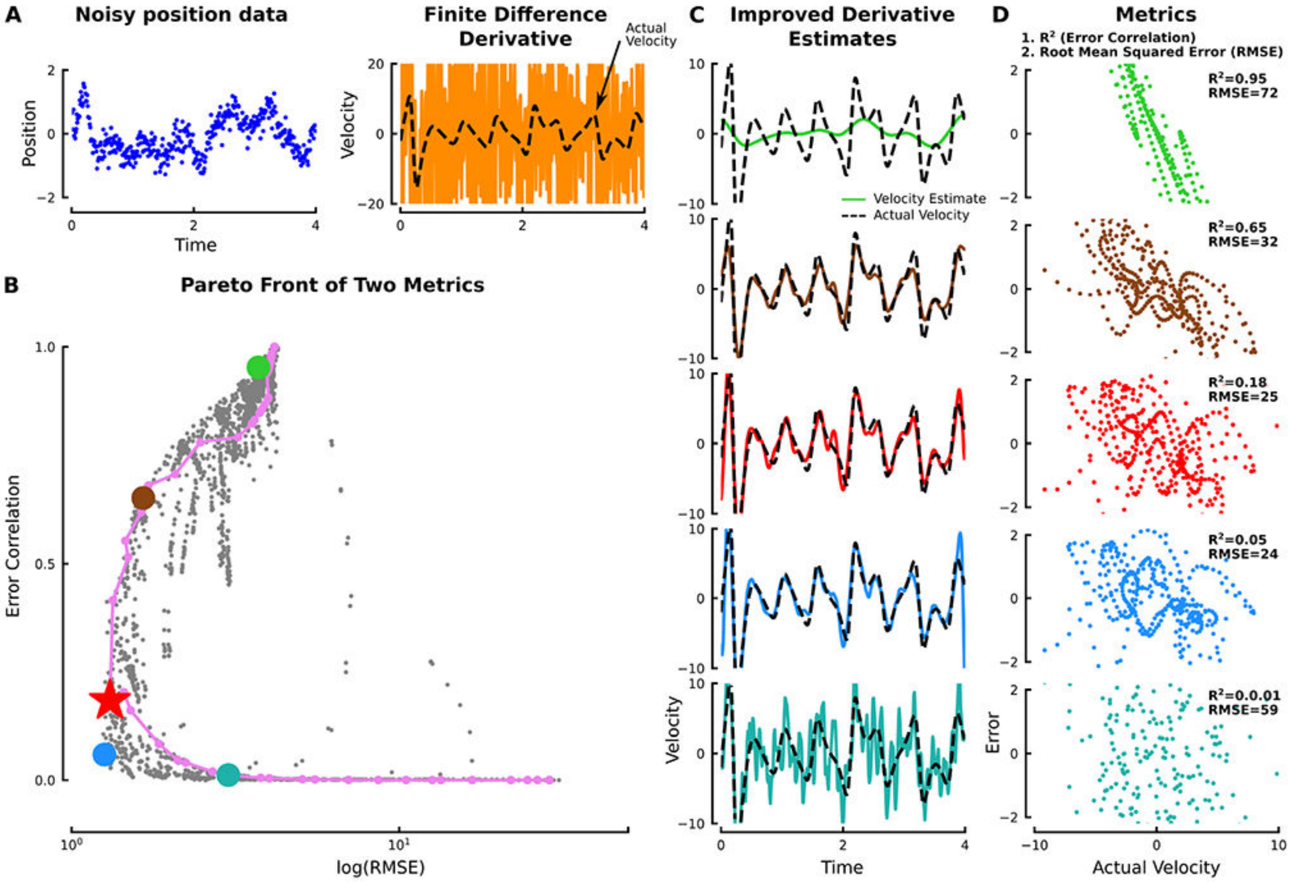

Fig. 1:

Choice of parameters leads to a diversity of derivative estimates. A. Noisy time series data, from a Lorenz system, and the corresponding finite difference derivative. B. To evaluate the quality of a derivative estimate relative to the ground truth, we consider two metrics: Root Mean Square Error (RMSE), and the Pearson's correlation coefficient (R^2) between the error and the true value of the derivative. Gray dots show the values of these metrics for 5,481 different sets of parameter choices for a smoothed Savitzky-Golay filter. The violet line is the result of our multi-objective optimization framework and nearly traces the Pareto front of the metrics. The derivative estimates and metrics for the five colored points along the Pareto front are shown in C and D, respectively.

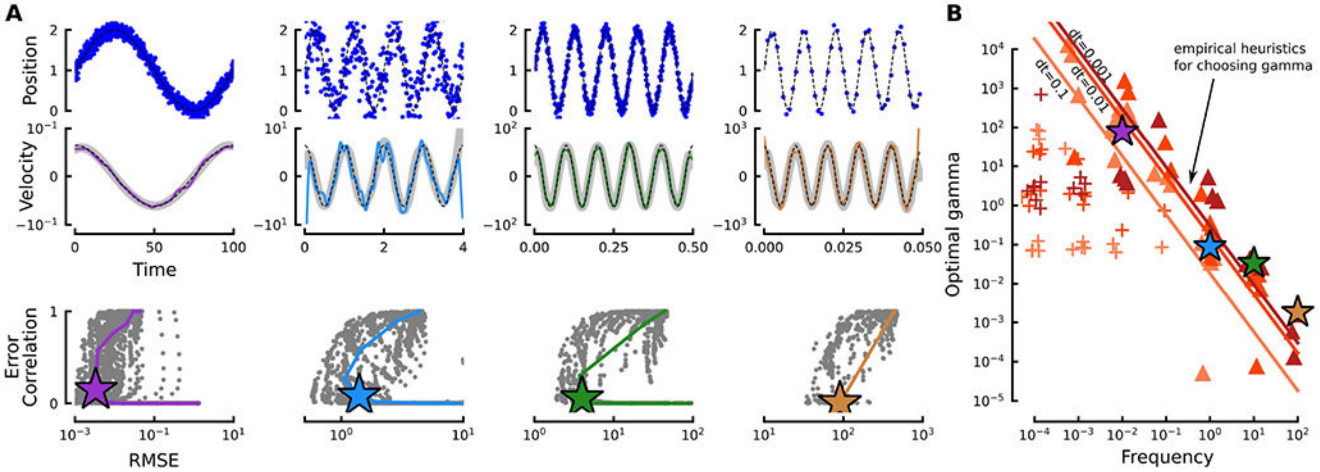

Fig. 2:

Optimal choice of γ is a function of frequency and temporal resolution of the data. A. (Top) Four example sine waves of different frequencies (note the time scales), temporal resolutions, and noise levels. (Middle) Comparison of the actual derivative (black dashed) with Savitzky-Golay estimates: lowest achievable RMSE (gray), and the result from our loss function with the optimal choice of γ defined in the bottom panel. (Bottom) Trade-off between error correlation and RMSE for 5,481 potential parameter choices (gray) and the options provided by our loss function for a sweep through γ (colored line). The star indicates the optimal choice of gamma, corresponding to the shoulder of the colored curve. (B) The optimal choice of γ (defined in A) as a function of frequency (Hz), for different temporal resolutions of data (0.001, 0.01, 0.1 sec). Also included in the plot, but not indicated, are different noise levels (0-mean normally distributed with standard deviations of 0.05%, 0.5%, 5%, and 25% of the amplitude) and length of the dataset (1, 4, 5, 25, 100, 500, 1000 sec). The ”+” markers indicate results from datasets for which the period was greater than the length of the time series, which were omitted from the fit. The diagonal lines indicate the empirical heuristics for choosing γ based on a multivariate ordinary least squares model, provided in Eqn. 8 and Table III.

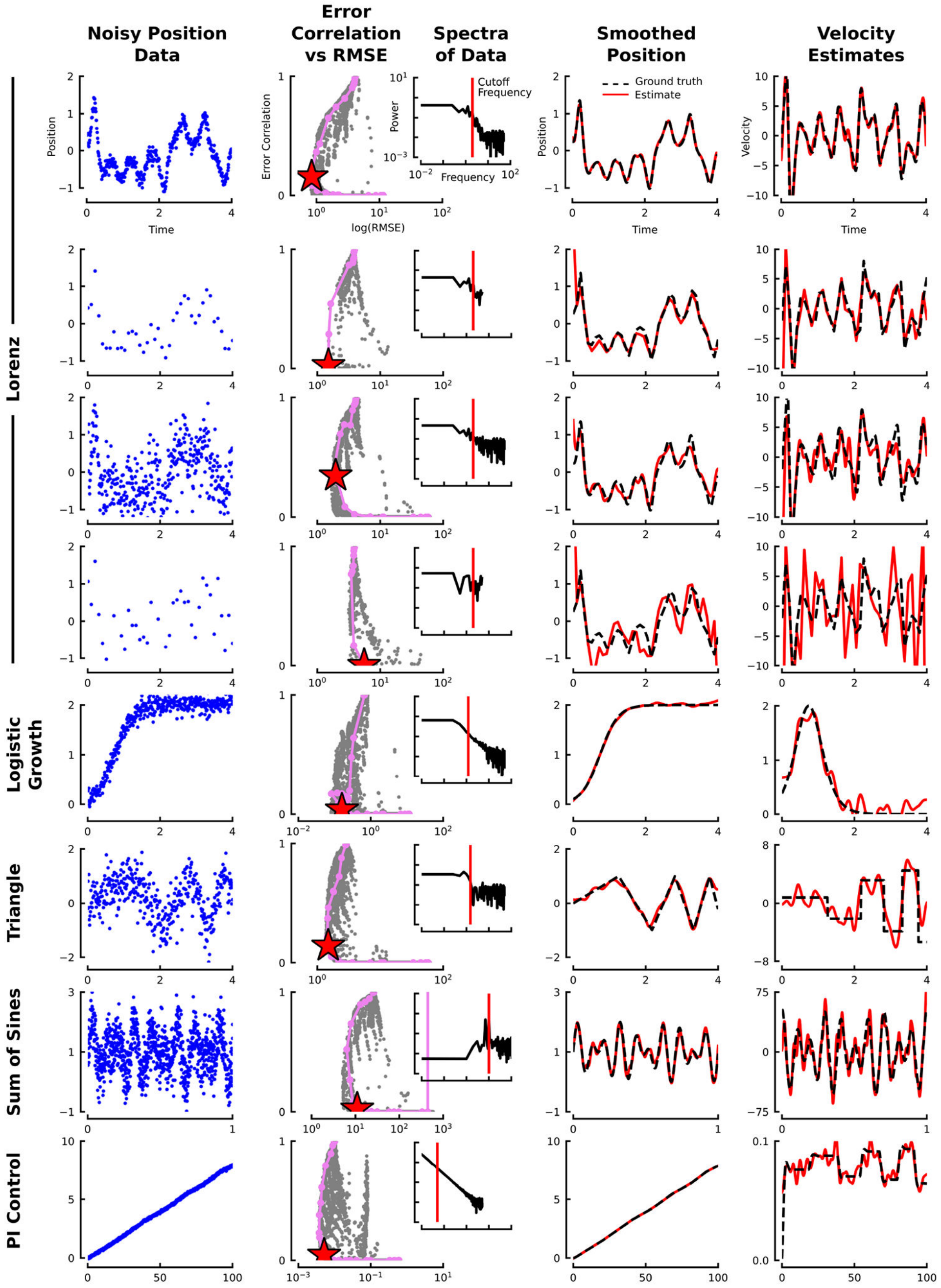

Fig. 3:

Heuristic for choosing γ is effective across a broad range of toy problems, using a Savitzky-Golay filter. The first column shows raw (synthetic) position data, indicating the shape of the data, degree of noise, and temporal resolution. Next we evaluate the performance of derivative estimate using the metrics described in the Fig. 1. Gray dots indicate the range of outcomes for 5,481 parameter choices, the violet line indicates the options provided by our loss function, and the red star indicates the performance using the suggested value of γ according to Eqn. 8. Frequency of the data is evaluated by inspecting the power spectra; the red line indicates the frequency used to determine γ. The final two columns compare the ground truth and estimates for position and velocity.

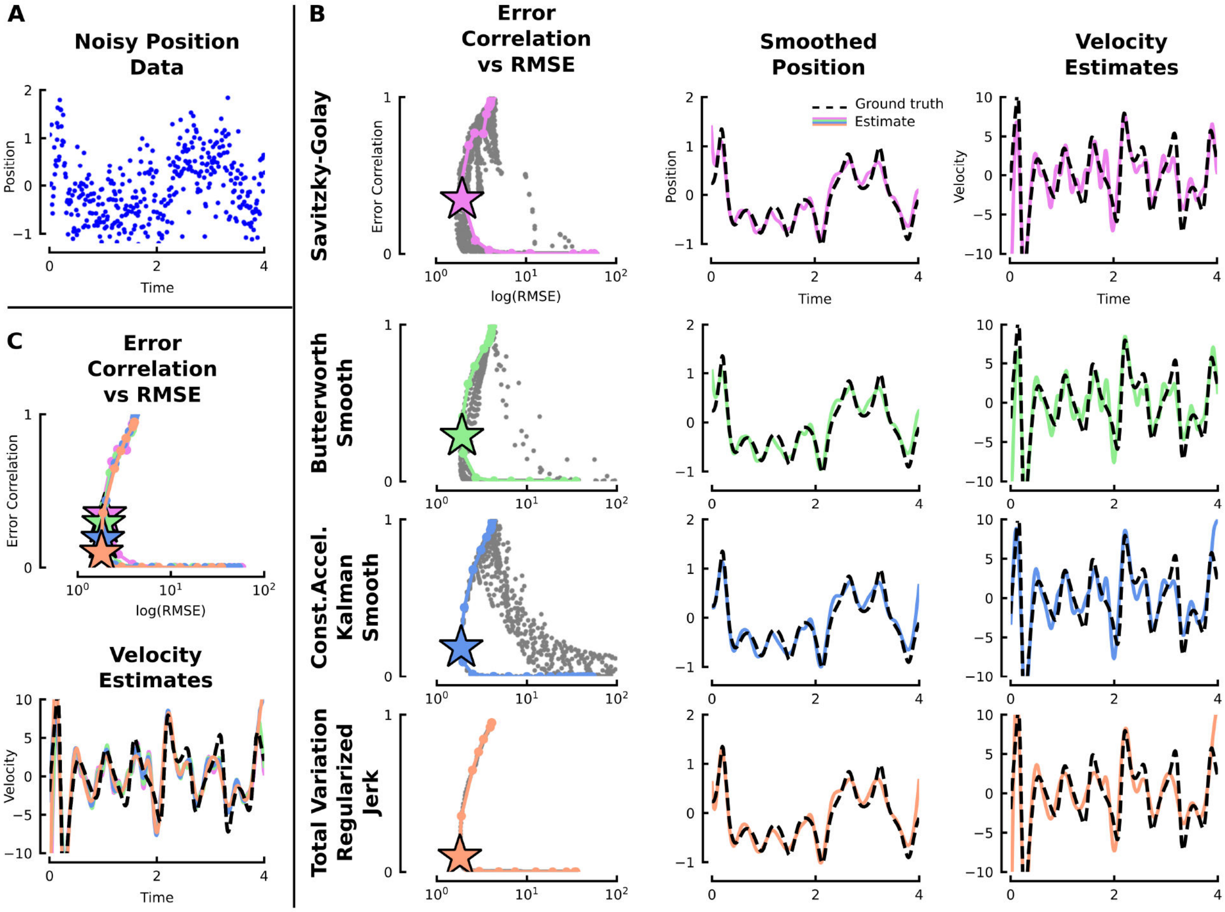

Fig. 4:

Loss function and heuristic for choosing γ is equally effective for different differentiation methods. A. Synthetic noisy data from the same Lorenz system as shown in Fig. 3. B. Comparison of metrics, position, and velocity estimates using four differentiation methods, with the same value of γ, as determined through the spectral analysis in Fig. 3. C. Overlay of the Pareto fronts and velocities for all four methods.

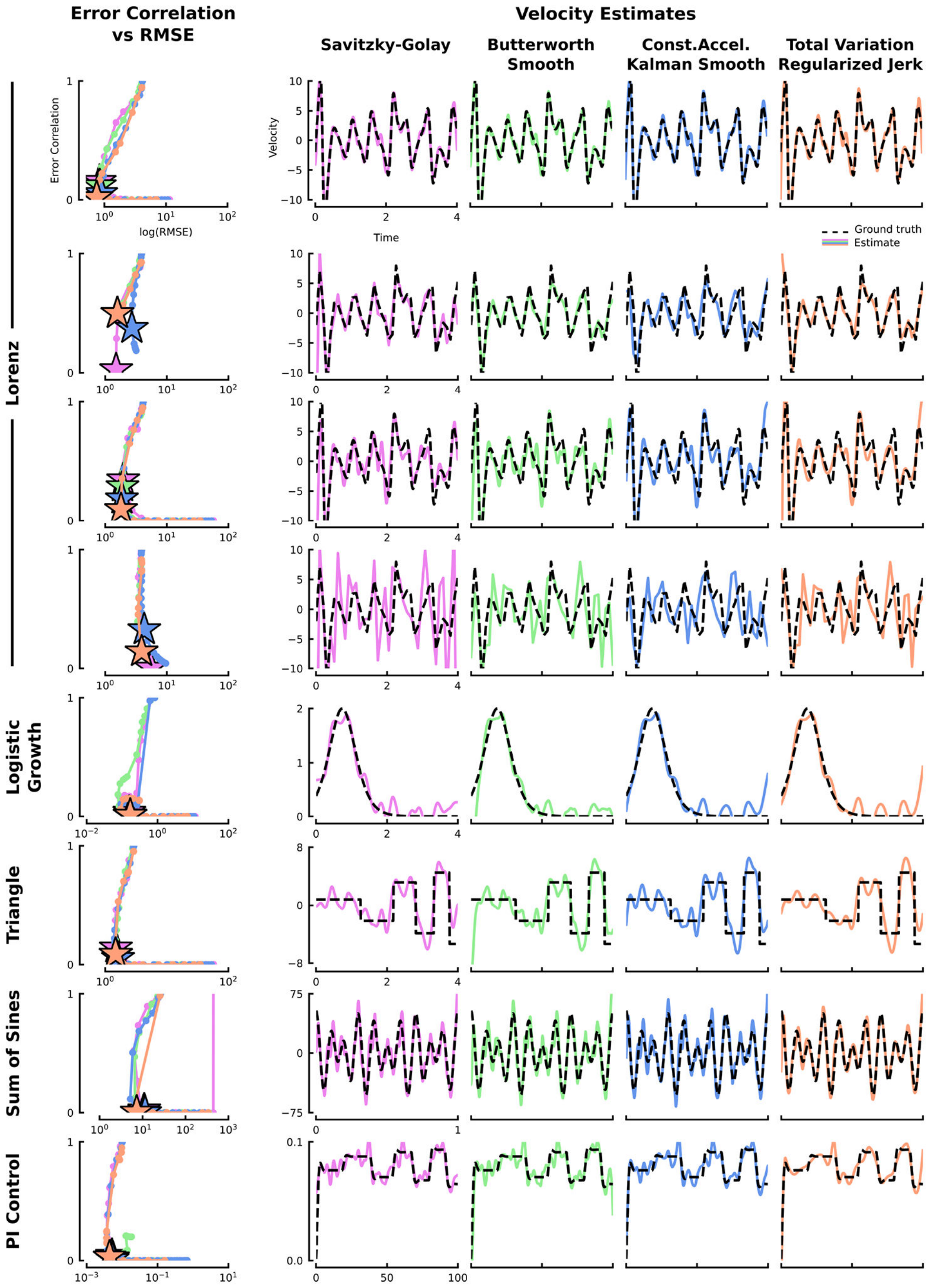

Fig. 5:

Loss function and heuristic for choosing γ is equally effective for different differentiation methods across a range of toy problems. Data plotted as in Fig. 4C, for each of the scenarios presented in Fig. 3.

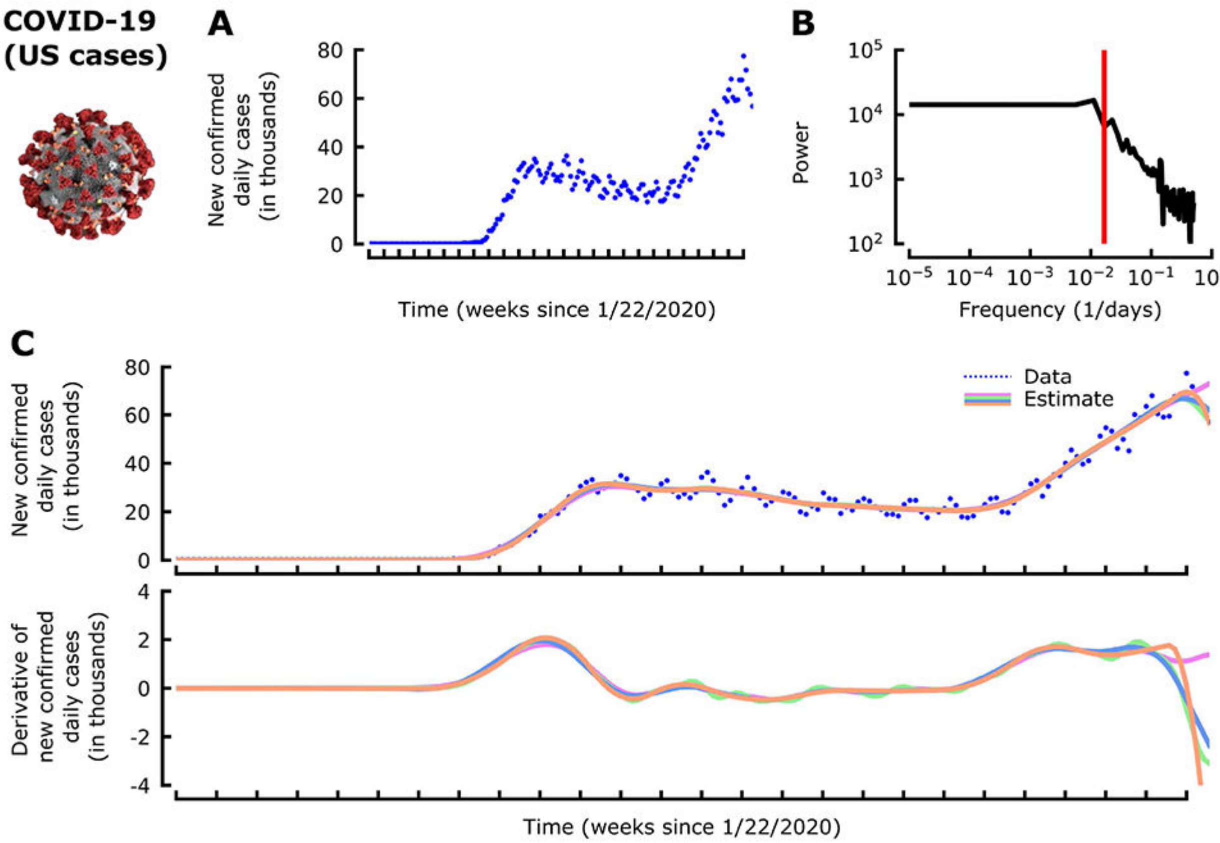

Fig. 6:

Numerical differentiation of new confirmed daily cases in the United States of COVID-19 [34] with no parameter tuning. A. Raw new daily cases. B. Power spectra of the data, indicating the cutoff frequency (red) used for selecting $\gamma = 4.1$. C. Smoothed new daily cases, and their derivative, using a Savitzky-Golay filter (violet), a Butterworth filter (green), a constant acceleration Kalman forward-backward smoother (light blue), and total variation regularized jerk (orange). Note the similarity between all four methods except in the very last week, despite the significant differences in how each method works and the automated parameter selection.

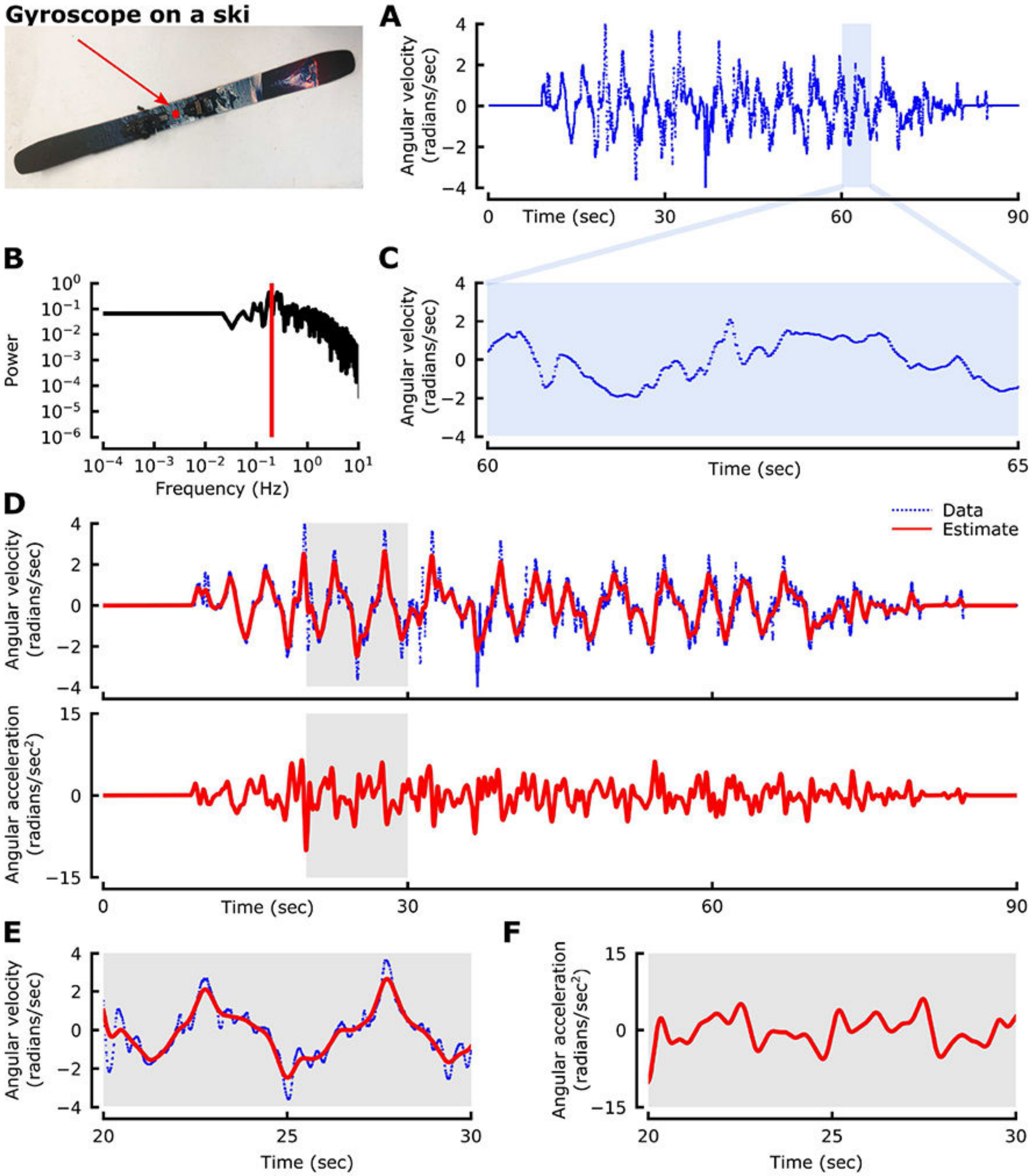

Fig. 7:

Numerical differentiation of noisy gyroscope data from a downhill ski during one ski run, with no parameter tuning. A. Data from one axis of a gyroscope attached to the center of a downhill ski. B. Power spectra of the data, indicating the cutoff frequency (red) used for selecting $\gamma = 11.5$. C. Zoomed in section of the data from A, which was used to optimize parameter selection. D. Smoothed angular velocities and angular accelerations, calculated using a Savitzky-Golay filter and the optimal parameters determined using our heuristic and loss function. E-F. Zoomed in sections from D.

TABLE I:

Summary of the four differentiation methods highlighted in this paper.

Full name	Abbreviated name	# Parameters	Computational cost	References
Butterworth filter followed by finite difference	Butterworth	2	low	[18]
Smooth Savitzky-Golay filter	Savitzky-Golay	3	low	[21], [22]
Constant acceleration forward-backward Kalman smoother	Kalman smooth	2	high	[26]
Total Variation Regularized Jerk	TVRJ	1	high	[28]

TABLE II:

Optimal log(γ) is correlated with frequency and temporal resolution, but not the noise or length of the dataset. The table provides the coefficients and associated p-values for a ordinary least squares model, with an adjusted $R^2 = 0.78$.

Variable	Coeff	P-value
intercept	−5.26	0
log(*freq*)	−1.55	0
log(*dt*)	−0.74	0
log(*noise*)	0.11	0.32
log(*length*)	0.10	0.32

TABLE III:

Optimal log(γ) can be determined based on the frequency and temporal resolution of the data. The table provides the coefficients and associated p-values for a ordinary least squares model, with an adjusted R^2 = 0.78.

Variable	Coeff	P-value
intercept	−5.1	0
log(*freq*)	−1.6	0
log(*dt*)	−0.71	0