Making Algebraic Parameter Estimation Practical for Noisy Data via Gaussian Process Regression

Oren Bassik
obassik@gradcenter.cuny.edu

November 12, 2025

Abstract

Contents

1 Introduction

Parameter estimation for ordinary differential equation (ODE) models is a fundamental challenge in nearly every scientific and engineering discipline. While ODEs provide a powerful language for describing the dynamics of physical, biological, and economic systems, their utility often hinges on determining the correct values for unknown parameters from experimental data. The dominant paradigm for this task relies on nonlinear optimization, often called a "shooting" or "simulation-based" approach. These methods iteratively guess parameter values, simulate the ODE, and adjust the guess to minimize the discrepancy between the simulation and the observed data.

While widely adopted, this optimization-based framework suffers from well-known practical limitations. Its performance is often highly sensitive to the initial parameter guesses, a value that is rarely known *a priori*. Consequently, these methods are susceptible to converging to local, non-optimal minima in the error landscape, potentially yielding incorrect parameter estimates. Furthermore, they are designed to find only a single set of optimal parameters, which is a significant drawback for systems that are only locally identifiable and may possess multiple, distinct parameter sets that explain the data equally well.

An alternative, the differential-algebraic approach, offers a compelling solution to these problems. By transforming the ODE system into a set of algebraic equations, this method eliminates the need for initial guesses and can, in principle, recover all valid parameter sets. This makes it a powerful tool for robust parameter estimation and for exploring the full landscape of system identifiability. However, the practical adoption of algebraic methods has been severely hampered by a critical flaw: an extreme sensitivity to measurement noise. The method relies on computing high-order derivatives of the observed data, a process that catastrophically amplifies even minuscule amounts of noise. This has relegated the algebraic approach to a theoretical curiosity, largely unusable for the noisy data characteristic of real-world experiments.

This paper bridges the gap between the theoretical promise of algebraic methods and the practical demands of real-world data. We introduce a novel approach that replaces the fragile derivative estimation step of the classic algebraic method with a robust, probabilistic smoother: Gaussian Process Regression (GPR). We demonstrate that this GPR-enhanced algebraic method preserves the key advantages of the original approach—no initial guesses, recovery of all solutions—while achieving a level of robustness to noise that makes it competitive with, and in some cases superior to, traditional optimization-based techniques.

Through a comprehensive benchmark on a diverse suite of 11 dynamical systems, we show that our method successfully estimates parameters in noise regimes where the original algebraic approach fails completely. This work demonstrates that by addressing the critical bottleneck of derivative estimation, we can finally make algebraic parameter estimation a practical, robust, and powerful tool for the modern scientist and engineer.

2 Background

To understand our contribution, it is necessary to first review the two core technologies it combines: the differential-algebraic approach to parameter estimation and Gaussian Process Regression for function approximation.

2.1 The Differential-Algebraic Approach

The algebraic approach to parameter estimation fundamentally reframes the problem from one of numerical optimization to one of algebraic root-finding. The core idea, as detailed in works such

as [?], is to eliminate the state variables from the system of ODEs through symbolic differentiation and substitution, resulting in a set of equations that relate the system parameters directly to the observed outputs and their higher-order derivatives.

Consider a general ODE system:

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{p}) \tag{1}$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{p}) \tag{2}$$

By repeatedly differentiating the output equation $\mathbf{y}(t)$ with respect to time and substituting in the expressions for $\mathbf{x}'(t)$ from the system dynamics, one can generate a tower of higher-order derivative equations. For a structurally identifiable system, this process eventually yields a set of polynomial equations involving only the parameters $\mathbf{p}$ and the time derivatives of the outputs, $\mathbf{y}, \mathbf{y}', \mathbf{y}'', \ldots$.

Once this symbolic manipulation is complete, the problem becomes concrete: if one can provide accurate numerical values for the derivatives of the outputs at a specific time point, the problem reduces to solving a system of polynomial equations for the unknown parameters $\mathbf{p}$. This approach carries two transformative advantages:

1. **No initial guesses are required.** Unlike optimization methods that traverse an error landscape, polynomial solvers are designed to find all solutions within a given domain directly.

2. **All solutions can be found.** For systems with multiple valid parameter sets (i.e., those that are only locally identifiable), this method can return the complete set of solutions, providing a full picture of the system's identifiability.

However, as noted, the practical utility of this powerful method hinges entirely on the ability to obtain accurate numerical estimates of the output derivatives from noisy data—a notoriously difficult task.

2.2 Gaussian Process Regression for Derivative Estimation

The challenge of accurately estimating derivatives from noisy data is the primary bottleneck for the algebraic method. Simple finite differences or interpolation schemes like polynomial or rational interpolation tend to overfit the noise, leading to wildly inaccurate and unstable derivative estimates. What is required is a method that can look past the noise to infer the most likely underlying smooth function that generated the data. This is a problem of regression, not interpolation.

Gaussian Process Regression (GPR) is a non-parametric, Bayesian method for regression that is exceptionally well-suited to this task [?]. Instead of fitting a single function to the data, GPR defines a prior distribution over a space of functions. When presented with data, it updates this prior to a posterior distribution that represents the most likely functions given the observations.

The key features of GPR that make it ideal for our purpose are:

- **Principled Smoothing:** A smoothness assumption is encoded in the prior via a kernel function (e.g., the squared exponential or RBF kernel). This allows the model to separate the underlying smooth signal from the high-frequency noise.

- **Explicit Noise Modeling:** GPR explicitly models the measurement noise as part of its generative model, typically as a Gaussian with a variance σ_n^2. Crucially, this noise variance, along with other kernel hyperparameters, can be learned directly from the data by maximizing the marginal likelihood.

- **Differentiable Mean Function:** The primary output of a trained GPR model is the posterior mean function, which is a smooth, infinitely differentiable function that represents the best estimate of the underlying signal. This function can be differentiated analytically or, more conveniently, via automatic differentiation to any desired order.

By using GPR, we are not merely smoothing the data; we are performing a principled Bayesian inference to find the most probable underlying function. As established in the comprehensive benchmark by [**?**], GPR stands out as the most robust and accurate method for estimating high-order derivatives from noisy time-series data. This finding provides the empirical justification for its use as the core of our robust algebraic parameter estimation framework.

3 Methodology

Our method synthesizes the theoretical strengths of the differential-algebraic approach with the practical robustness of Gaussian Process Regression. The core innovation is the replacement of a noise-sensitive interpolation step with a principled, probabilistic regression that yields stable, high-order derivatives. The overall workflow can be broken down into five key steps, which we detail below.

3.1 Algorithm Overview

1. **Symbolic Differentiation:** The process begins with a purely symbolic analysis of the ODE system. Following the differential-algebraic methodology, we repeatedly differentiate the system's output equations with respect to time, substituting the state dynamics at each step. This procedure generates a system of equations relating the system parameters $\mathbf{p}$ to the time derivatives of the observable outputs $\mathbf{y}, \mathbf{y}', \mathbf{y}'', \ldots, \mathbf{y}^{(d)}$. The required order of differentiation, d, is determined by the structure of the model and can often be determined by identifiability analysis software.

2. **GPR-based Derivative Estimation:** This step is the central contribution of our work. Given a time series of noisy measurements $\{(t_i, y_i)\}$, we fit a Gaussian Process Regression model to the data for each output variable. From the trained GPR model, we extract the posterior mean function, $\mu_{GP}(t)$, which represents the most probable underlying smooth function that generated the data. We then use automatic differentiation (specifically, Taylor-mode AD for efficiency) to compute the numerical values of this function's derivatives, $\mu_{GP}(t_0), \mu'_{GP}(t_0), \ldots, \mu^{(d)}_{GP}(t_0)$, at a chosen time point t_0 (typically $t_0 = 0$).

3. **Polynomial System Assembly:** The numerical derivative values obtained from the GPR models are substituted into the symbolic equations derived in Step 1. This transforms the differential system into a concrete, numerical system of polynomial equations where the only unknowns are the parameters $\mathbf{p}$ and the state variables $\mathbf{x}(t_0)$ at the chosen time point.

4. **Numerical Root Finding:** The resulting polynomial system is solved using a numerical algebraic geometry solver. We primarily use homotopy continuation, a robust technique for finding all complex solutions to a system of polynomial equations. The real-valued solutions from this set form our candidate parameter sets.

5. **Solution Filtering and Validation:** Each candidate parameter set $(\hat{\mathbf{p}}, \hat{\mathbf{x}}_0)$ is used to simulate the original ODE system over the full time domain. The root-mean-square error (RMSE)

between each simulation and the original measurement data is computed. The parameter set yielding the lowest RMSE is selected as the final estimate. For systems that are only locally identifiable, this step may yield multiple distinct parameter sets with similarly low error, thereby revealing the system's identifiability structure.

3.2 GPR Implementation for Robust Differentiation

The success of this entire pipeline hinges on the quality of the derivative estimates in Step 2. As established in our companion benchmark study [**?**], the choice of GPR implementation is critical. Our approach uses the following configuration:

- **Kernel:** We employ a squared exponential (also known as Radial Basis Function or RBF) kernel, which embeds the prior assumption that the underlying function is smooth and infinitely differentiable—a natural fit for the solutions of most physical ODE systems.

- **Hyperparameter Optimization:** The GPR model's hyperparameters, including the kernel's length scale and the variance of the measurement noise, are not set manually. Instead, they are automatically learned from the data by maximizing the log marginal likelihood. This allows the model to adaptively determine the appropriate level of smoothing based on the observed data.

- **Automatic Differentiation:** After fitting, we do not use kernel-specific derivative formulas. We extract the posterior mean function and apply Taylor-mode automatic differentiation [**?**] to it. This provides an efficient and numerically exact method for computing all required high-order derivatives in a single pass, avoiding the exponential complexity associated with repeated application of forward- or reverse-mode AD.

By combining these elements, we create a derivative estimation engine that is not only robust to noise but also highly automated, requiring no manual tuning of smoothing parameters. This "fit-then-differentiate" paradigm, powered by GPR and Taylor-mode AD, is the key enabling technology that makes the algebraic method practical for real-world applications.

4 Experimental Setup

To rigorously evaluate the performance and robustness of our GPR-enhanced algebraic method, we designed a comprehensive experimental protocol using a diverse suite of benchmark systems. This section details the systems under study, the data generation process, the baseline methods used for comparison, and the metrics for evaluation.

4.1 Benchmark Systems

We selected 11 ODE models from various scientific domains to ensure our evaluation is not biased toward a specific type of system dynamics. The suite includes classic models from population dynamics, epidemiology, immunology, and neuroscience, representing a wide range of complexities in terms of the number of states and parameters. A full description of these systems can be found in the appendix of [**?**].

4.2 Data Generation and Noise Protocol

For each benchmark system, we generated multiple datasets to test performance under various conditions.

- **True Parameters:** For each of 10 trials, true parameter values were drawn randomly from a uniform distribution $U[0.1, 0.9]$.
- **Data Simulation:** The ODE system was solved using a high-precision numerical integrator to generate a baseline, noise-free trajectory. Data was sampled at 100 equally spaced time points.
- **Noise Injection:** To simulate realistic experimental data, we added Gaussian white noise to the clean trajectory. We tested performance across a wide spectrum of five noise levels, corresponding to 0%, 0.01%, 0.1%, 1%, and 2% of the standard deviation of the noise-free signal. This resulted in 11 systems $\times$ 10 trials $\times$ 5 noise levels = 550 unique datasets.

4.3 Baseline Methods for Comparison

To contextualize the performance of our method, we compare it against two classes of estimators: the original, noise-sensitive algebraic method and a representative state-of-the-art optimization-based method.

- **ODEPE (AAA):** This represents the original algebraic method that uses the Adaptive Antoulas-Anderson (AAA) rational interpolation algorithm for derivative estimation. This baseline serves to demonstrate the impact of noise on non-regressive approaches.
- **SciML:** This represents a modern, highly-optimized shooting method implemented in the Julia SciML ecosystem [**?**]. It uses a gradient-based local optimization algorithm (a combination of ADAM and L-BFGS) to find a single best-fit parameter set. This serves as a strong baseline representing the standard approach in the field.
- **Our Method (ODEPE-GPR):** This is the GPR-enhanced algebraic method proposed in this paper.

4.4 Evaluation Metrics

We evaluate the methods based on two primary metrics that capture both the accuracy and reliability of the parameter estimates.

- **Success Rate:** An estimation run is considered a "success" if the mean relative error of all estimated parameters is less than 10%. The success rate is the percentage of the 10 trials for a given system and noise level that meet this criterion. This metric measures the reliability and robustness of a method.
- **Median Relative Error:** For the successful runs, we compute the median of the mean relative errors. We use the median because it is robust to occasional outlier runs where the method may have found a particularly poor local minimum, providing a better sense of the method's typical accuracy.

Together, these metrics provide a comprehensive picture of a method's real-world performance: its likelihood of finding a reasonable solution (Success Rate) and the quality of that solution when it does (Median Relative Error).

5 Results

Our experimental evaluation clearly demonstrates the transformative impact of replacing interpolation with Gaussian Process Regression in the algebraic parameter estimation pipeline. The GPR-enhanced method not only proves robust to significant measurement noise but also consistently matches or exceeds the performance of a state-of-the-art optimization-based approach.

5.1 Overall Performance

We first assess the overall performance of the three methods—our GPR-enhanced algebraic method (ODEPE-GPR), the optimization-based SciML, and the original interpolation-based algebraic method (ODEPE-AAA)—averaged across all 11 benchmark systems and all noise levels. The results are summarized in Table 1.

Table 1: Overall Performance Comparison Across All Systems and Noise Levels

Method	Success Rate (%)	Median Rel. Error
ODEPE-GPR (Ours)	99.6	0.003
SciML (Optimization)	99.6	0.004
ODEPE-AAA (Interpolation)	38.2	0.003

As the table shows, both ODEPE-GPR and SciML achieve a near-perfect success rate of 99.6%, indicating that they are both highly reliable at finding accurate parameter estimates across a wide range of conditions. In stark contrast, the original ODEPE-AAA method has a success rate of only 38.2%. This confirms our central hypothesis: the reliance on interpolation renders the traditional algebraic method brittle and unreliable in the presence of noise. While its median error on the few successful runs is low, it fails to find a correct solution in the majority of cases. Our GPR-based method, however, is just as reliable as the established, optimization-based SciML approach.

5.2 Robustness to Noise

To understand the source of this performance difference, we analyze the median relative error of each method as a function of the noise level. The results, presented in Table 2, are striking.

Table 2: Median Relative Error by Noise Level

Noise Level	ODEPE-GPR (Ours)	SciML (Optimization)	ODEPE-AAA (Interpolation)
0.0%	0.000	0.000	0.000
0.01%	0.003	0.003	0.003
0.1%	0.030	0.030	2.459
1.0%	0.288	0.292	23.32
2.0%	0.548	0.582	43.83

At zero and very low (0.01%) noise levels, all three methods perform identically and achieve near-perfect accuracy. This is expected, as the algebraic method is known to be highly effective on clean data. However, the performance diverges dramatically as the noise increases.

At a moderate noise level of 0.1%, the error of the ODEPE-AAA method explodes, increasing by nearly three orders of magnitude. By contrast, our ODEPE-GPR method's error increases

gracefully, remaining on par with the SciML optimization baseline. At high noise levels (1% and 2%), the ODEPE-AAA method fails completely, with errors that are orders of magnitude larger than the other methods. Remarkably, our ODEPE-GPR method continues to track the performance of SciML closely, demonstrating a robust and predictable degradation of performance as the task becomes more difficult.

These results tell a clear story: the GPR-enhancement successfully tames the noise sensitivity of the algebraic approach. It makes the method robust, allowing it to perform just as well as a leading optimization-based method in low-to-moderate noise regimes, where the original algebraic method is entirely unusable.

6 Discussion

The results presented in the previous section confirm that the integration of Gaussian Process Regression fundamentally transforms the practicality of the differential-algebraic approach to parameter estimation. Here, we discuss the implications of these findings, the reasons for GPR's success, and the inherent limitations of the method.

6.1 Why the GPR-Enhanced Method Succeeds

The dramatic improvement in performance over the original AAA-based algebraic method can be attributed to the fundamental difference between interpolation and regression. Interpolation methods, by definition, are forced to honor the data points exactly. When data is noisy, an interpolant will necessarily weave through the scattered points, leading to a highly oscillatory function whose derivatives are wildly inaccurate.

GPR, on the other hand, operates under a different paradigm. It assumes that the data is generated by a smooth, underlying function corrupted by noise. Instead of fitting the noise, GPR uses Bayesian inference to find the most likely smooth function given the data. This is achieved through two key mechanisms:

1. **A Principled Smoothness Prior:** The choice of a smooth kernel (like the RBF kernel) acts as a "regularizer," penalizing functions that are not smooth and guiding the model to find a simpler, more plausible underlying signal.

2. **Explicit Noise Modeling:** GPR does not assume the data is perfect. It explicitly models a noise term, and, crucially, learns the magnitude of this noise directly from the data. This allows it to automatically determine how much to trust the data points, smoothing more aggressively when the data is noisy and fitting more closely when the data is clean.

By treating derivative estimation as a regression problem rather than an interpolation problem, we directly address the core weakness of the algebraic method, enabling it to function effectively in realistic, noisy conditions.

6.2 Practical Implications for Researchers

The successful robustification of the algebraic method has significant practical implications for researchers and engineers who build and use ODE models:

- **Elimination of Initial Guesses:** The most immediate benefit is the removal of the need for manual tuning of initial parameter guesses. This lowers the barrier to entry for complex models and makes the estimation process more systematic and reproducible.

- **Built-in Identifiability Analysis:** Optimization methods typically return a single point estimate, giving no information about other possible solutions. Our method, by virtue of its root-finding approach, can return all possible parameter sets consistent with the data. This provides invaluable insight into the structural identifiability of a model, a feature often overlooked in standard parameter fitting workflows.
- **A New Tool for Model Exploration:** The GPR-enhanced algebraic method can be used as a standalone estimation tool, especially in low-to-moderate noise regimes. Alternatively, it can serve as a powerful initialization method for traditional optimization-based approaches. By providing a high-quality, data-driven starting point (or multiple starting points in the case of local identifiability), it can significantly improve the efficiency and reliability of subsequent local optimization.

In essence, this work makes the powerful theoretical advantages of the algebraic approach available as a practical tool for everyday modeling work.

6.3 Limitations and Future Work

Despite its strong performance, our method is not without limitations.

- **Computational Cost:** The primary bottleneck is the computational complexity of both GPR and the polynomial solver. Standard GPR scales as $O(N^3)$ with the number of data points N, which can become prohibitive for very large datasets. Similarly, the difficulty of solving a polynomial system grows rapidly with the number of parameters and the degree of the equations. Future work should explore the use of sparse GPR approximations to improve scaling for large data.
- **Data Requirements:** As a two-stage method, our approach requires data that is sufficiently dense to allow for a reasonable regression fit. It is likely less suitable for extremely sparse data, where simulation-based methods that can integrate between points may have an advantage.
- **Reliance on Solvers:** The final step of the method depends on the successful convergence of a numerical polynomial solver. While modern homotopy continuation methods are very robust, they are not infallible and can struggle with poorly conditioned or very large systems.

Future research will also focus on propagating the uncertainty estimates from the GPR model through the entire pipeline, which could provide confidence intervals on the final parameter estimates—a feature currently lacking in our framework.

7 Conclusion

References