Robust Algebraic Parameter Estimation via Gaussian Process Regression

Oren Bassik[1] Alexander Demin[2] Alexey Ovchinnikov[1,3]

IFAC/COSY/SSSC Joint Conference

July 2, 2025

[1]CUNY Graduate Center
[2]HSE University
[3]CUNY Queens College

Supported by NSF grants CCF-2212460 and DMS-1853650

Robust Differentiation: The Cornerstone of Algebraic Parameter Estimation

Challenge: Differentiating Noisy Data

- Algebraic estimators require derivatives $y(t), \dot{y}(t), \ddot{y}(t), \ldots$ from noisy measurements $y_k = y(t_k) + \varepsilon_k$
- Naive differentiation acts as a high-pass filter
- Fundamental issue: $\mathrm{Var}[\Delta^n y / \Delta t^n] \propto \sigma^2 \Delta t^{-2n}$
- **Result:** Significant noise amplification, degrading derivative quality

TODO: Plot showing AAA interpolant
oscillating wildly through noisy data
with corresponding unstable derivative

GPR Approach: Probabilistic Differentiation

- Reframe as Bayesian inference: $y(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$
- Smoothness prior via kernel (RBF, Matérn) encodes physical assumptions
- Derivatives computed analytically: $\partial^\ell y | \mathcal{D} \sim \mathcal{GP}(\partial^\ell \mu(t), \partial^\ell \partial^{\ell'} \Sigma(t, t'))$
- **Result:** Stable derivatives with uncertainty quantification

TODO: Plot showing smooth GPR mean
with confidence bands and
corresponding stable derivative

Key Insight

Supported by NSF grants CCF-2212460 and DMS-1853650

Evaluating Differentiator Performance

Noise Sensitivity

RMSE vs. Noise Level

TODO: Log-log plot showing:

- *GPR_Julia: stable slope $\alpha \approx 1.1$*
- *AAA: catastrophic slope $\alpha \approx 9.7$ at $\sigma > 10^{-8}$*
- *95% confidence intervals*
- *Vertical line at failure threshold*

Higher-Order Derivatives

RMSE vs. Derivative Order

TODO: Semi-log plot showing:

- *GPR_Julia: graceful degradation*
- *TVDiff: complete failure at order ≥ 4*
- *Finite differences: exponential growth*
- *Failure markers for non-convergent cases*

Noise Cliff Analysis

- **GPR_Julia:** Stable performance across all noise levels
- **AAA methods:** High error sensitivity above $\sigma \approx 10^{-8}$
- **Error growth:** GPR error grows linearly; AAA error grows exponentially
- **Statistical sig:** $p < 10^{-12}$ for difference at $\sigma = 10^{-6}$

Derivative Order Analysis

- **GPR_Julia:** Graceful degradation with increasing order
- **TVDiff:** Fails to converge for orders ≥ 4
- **Finite differences:** Rapid error growth

Supported by NSF grants CCF-2212460 and DMS-1853650

Why Gaussian Process Regression Excels at Numerical Differentiation

Bayesian Framework

1. Prior over signal: $f(t) \sim \mathcal{GP}(0, k(t, t'))$

RBF kernel: $k(t, t'; \theta) = \sigma_f^2 \exp\left(-\frac{(t-t')^2}{2\ell^2}\right)$

2. Posterior inference: $y = f + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_n^2 I)$

$\mathbb{E}[f(t^*)|y] = k_*^T (K + \sigma_n^2 I)^{-1} y$

3. Analytical derivatives: $\frac{\partial f}{\partial t} \sim \mathcal{GP}(\cdot, \cdot)$

$\mathbb{E}[\dot{f}(t^*)|y] = {k'_*}^T (K + \sigma_n^2 I)^{-1} y$

4. Hyperparameter optimization:

$\log p(y|\theta) = -\frac{1}{2} y^T \Sigma^{-1} y - \frac{1}{2} \log |\Sigma| - \frac{n}{2} \log 2\pi$

TODO: GPR illustration showing:

- *Noisy data points*
- *GP posterior mean (smooth)*

Key Technical Advantages

1. **Probabilistic Function Inference**
   - Models underlying function $f(t)$, not just data
   - Treats differentiation as statistical inference
2. **Data-Driven Hyperparameters**
   - Marginal likelihood automatically finds $\{\sigma_f, \ell, \sigma_n\}$
   - No ad-hoc filter tuning required
3. **Numerical Stability**
   - Avoids $\Delta y / \Delta t$ noise amplification
   - Regularized kernel matrix: $(K + \sigma_n^2 I)^{-1}$
4. **Global Smoothing + Local Accuracy**
   - All data points inform each estimate
   - Kernel decay: exponential locality, matrix coupling: global

Supported by NSF grants CCF-2212460 and DMS-1853650

Application: Making the Algebraic Method Viable

Benchmarking Against Standard Methods

Test Suite: Nonlinear dynamic systems with realistic noise (1.0% rel. noise)

System	GPR-Algebraic	Original (AAA)	SciML (LM)
Fitzhugh-Nagumo	**1.8%**	¿100%	0.5%
Lotka-Volterra	**2.1%**	¿100%	0.3%
SEIR Model	**1.2%**	FAIL	0.4%
HIV Dynamics	**3.4%**	¿100%	0.5%

Errors are Mean Relative Error (MRE) in parameter estimates.
SciML is a standard Levenberg-Marquardt optimization solver.

TODO: Trajectory comparison plot

- *Ground truth vs. estimates*
- *GPR-Algebraic (green, accurate)*
- *Original AAA (red, diverging)*
- *Noisy measurements (gray dots)*

Key Contributions

- **Enables Robustness**
  - GPR component overcomes the brittleness of the original algebraic method.
- **Preserves Automation**
  - Retains the key algebraic advantage: no initial parameter guesses required.
- **Achieves Viability**
  - Performance is now competitive with established optimization-based methods.
- **Offers Alternate Approach**

Supported by NSF grants CCF-2212460 and DMS-1853650

Conclusions & Outlook

Technical Contributions

- **Solved key bottleneck** for algebraic method via a principled GPR approach
- **Demonstrated noise tolerance**
  - GPR-based method is effective up to $\sigma^2 \approx 10^{-2}$
  - Interpolation-based methods fail above $\sigma^2 \approx 10^{-8}$
- **Preserved algebraic advantages**
  - No initial parameter guesses
  - Fully automated operation
  - Analytic solution guarantees
- **Minimal computational overhead**
  - $<$ 20% extra CPU (100ms $\longrightarrow$ 118ms)
  - Polynomial solving remains bottleneck
- **Identified common failure modes**
  - "Noise cliff": high sensitivity

Scientific Contributions

- **Systematic evaluation framework**
  - 21 numerical differentiation methods
  - Unified Automatic Differentiation harness
  - Fair, reproducible comparisons
- **Identified universal failure modes**
  - "Noise cliff": catastrophic failure at critical σ
  - "Derivative wall": breakdown at high orders ($k \geq 4$)
- **Mathematical foundation**
  - Proved GP kernel smoothness $\Rightarrow$ well-posed algebraic estimator
  - Explains unique robustness of GP solution
- **Bridges algebraic & Bayesian methods**

Future Research Directions

- **From Estimates to Confidence**
  - Uncertainty quantification via GP posterior variance
  - Credible intervals on parameter estimates
  - Risk-aware parameter estimation
- **Data-Driven Experiment Design**
  - Optimal experimental design using variance field
  - A- and D-optimality for parameter estimation
  - Active learning for minimal data collection
- **Scaling the Framework**
  - Sparse/inducing-point GPs $\Rightarrow$ $\mathcal{O}(N \log N)$
  - Extension to 10^2-10^3 parameters
  - Real-time implementation

Supported by NSF grants CCF-2212460 and DMS-1853650

A1: Computational Performance Analysis

Timing Benchmarks

Runtime Comparison (10k samples):

- GPR_Julia: 2.15s
- SavitzkyGolay: 0.0015s
- TVDiff: 0.19s
- FiniteDiff: 0.0006s

Runtime Breakdown (GPR):

- Polynomial solving: 70%
- GPR computation: 20%
- Miscellaneous: 10%

Scaling Analysis

Computational Complexity:

- Naive GP: $\mathcal{O}(N^3)$
- Sparse GP: $\mathcal{O}(NM^2)$, $M \ll N$

TODO: Bar chart showing runtime breakdown and method comparison

Key Insight

GPR is **not** the computational bottleneck. Accuracy-first approach justified by minimal overhead vs. dramatic robustness gains.

Supported by NSF grants CCF-2212460 and DMS-1853650

A2: Extended Benchmark Results

Complete Performance Matrix: 21 Methods $\times$ 3 Noise Levels $\times$ 6 Derivative Orders

TODO: Comprehensive heatmap showing

$\log_{10}$*(RMSE) across all conditions*

Red = failure, Green = success

Generated by method_comparison_heatmap.pdf

Statistical Analysis

- **Failure Rate Analysis:** GPR_Julia: 0% (0/63), AAA_Julia: 37% (23/63), TVDiff: 52% (33/63)
- **Edge Cases:** Very low noise ($\sigma < 10^{-10}$) requires numerical conditioning
- **Confidence Intervals:** 95% CI on geometric mean RMSE across 30 MC runs per condition
- **Method Categories:** Finite differences, spectral, variational, kernel-based

Supported by NSF grants CCF-2212460 and DMS-1853650

A3: GPR Implementation & Methodology Details

Hyperparameter Optimization

Marginal Likelihood Maximization:

$$\log p(y|\theta) = -\frac{1}{2} y^T \Sigma^{-1} y - \frac{1}{2} \log |\Sigma| - \frac{n}{2} \log 2\pi$$

Optimization Strategy:

- L-BFGS-B with multiple random initializations
- Parameter bounds: $\ell \in [10^{-3}, 10^3]$, $\sigma_f^2 \in [10^{-6}, 10^6]$
- Automatic noise floor: $\sigma_n^2 \geq 10^{-12}$

Kernel Selection

RBF vs. Matérn Comparison:

- RBF: C^∞ smooth, best for high-order derivatives
- Matérn-3/2: C^1 smooth, computational efficiency
- Matérn-5/2: C^2 smooth, good compromise

Integration Framework

Automatic Differentiation:

- Julia: ForwardDiff.jl for exact derivatives
- Python: JAX for vectorized operations
- Enables fair, consistent method comparison

Software Integration:

- Modular design: derivative estimator + algebraic solver
- Compatible with existing parameter estimation pipelines
- Export to MATLAB, Python, Julia formats

TODO: Workflow diagram

Data $\rightarrow$ GPR $\rightarrow$ Derivatives $\rightarrow$

Algebraic Solver $\rightarrow$ Parameters

Supported by NSF grants CCF-2212460 and DMS-1853650