

Bayesian Integral-based Methods for Differential Equation Models

by

Mingwei Xu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2025

© Mingwei Xu 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: David A. Campbell
 Professor, Carleton University

Supervisor(s): Samuel W.K. Wong
 Associate Professor, University of Waterloo
 Peijun Sang
 Associate Professor, University of Waterloo

Internal Member(s): Martin Lysy
 Associate Professor, University of Waterloo
 Alex Stringer
 Assistant Professor, University of Waterloo

Internal-External Member: Brian Ingalls
 Professor, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Mingwei Xu was the sole author for Chapters 1, 4 and 5 which were written under the supervision of Samuel W.K. Wong and Peijun Sang and have not been submitted for publication.

This thesis consists in part of two manuscripts written for publication. Exceptions to sole authorship of material are as follows:

Research presented in Chapter 2:

Xu, M., S. W. K. Wong, and P. Sang (2024). A Bayesian collocation integral method for parameter estimation in ordinary differential equations. *Journal of Computational and Graphical Statistics*, 33, 845–854. DOI: 10.1080/10618600.2024.2302528.

Research presented in Chapter 3:

Xu, M., S. W. K. Wong, and P. Sang (2025). A Bayesian collocation integral method for system identification in ordinary differential equations. Manuscript submitted for publication.

As lead author of these two chapters, I was responsible for contributing to conceptualizing methodology, carrying out simulation and real data analysis, and drafting manuscripts. My coauthors provided guidance during each step of the research and provided feedback on draft manuscripts.

Abstract

Ordinary differential equations (ODEs) are widely considered for modeling the dynamics of complex systems across various scientific areas. When inferring the parameters of ODEs from noisy observations, most existing parameter estimation methods that bypass numerical integration tend to rely on basis functions or Gaussian processes to approximate the ODE solution and its derivatives. Due to the sensitivity of the ODE solution to its derivatives, these methods can be hindered by estimation error, especially when only sparse time-course observations are available. Furthermore, in high-dimensional sparse ODE systems, existing methods for structure identification are predominantly frequentist, and uncertainty quantification for trajectory estimation remains a significant challenge.

In this thesis, we present a Bayesian collocation framework that operates on the integrated form of the ODEs and also avoids the expensive use of numerical solvers. In contrast to frequentist methods, the Bayesian framework provides better quantification of uncertainty. Specifically, we propose three methods to recover ODE systems with both known and unknown functional forms, as well as for model validation in systems with known functions. First, for ODEs with known functional forms, our methodology can handle general nonlinear systems to infer parameters from noisy observations. We then extend the approach to ODEs with unknown functional forms to identify system structure. Under the additive ODE model assumption, we develop a unified framework that combines the likelihood, integrated ODE constraints, and a group-wise sparse penalty, enabling simultaneous system identification and trajectory estimation. Finally, for cases involving several competing functional forms that exhibit similar dynamics, we incorporate model inadequacy as a function into the measurement model, allowing simultaneous model validation and parameter estimation. We demonstrate the favourable performance of our proposed methods, compared to existing ones, in terms of estimation accuracy and model validation. We also illustrate the methods with real data analyses.

Acknowledgements

I would like to thank my supervisors, Samuel Wong and Peijun Sang for their insights and guidance. Their dedication to both research and mentorship has profoundly shaped my academic development. Moreover, their example of maintaining a healthy work–life balance has been instrumental in helping me navigate the challenges of harmonizing family life with the demands of research.

I also thank Alex Stringer, Martin Lysy, Brian Ingalls, and David Campbell for serving as members of my examining committee. Their thoughtful insights and constructive suggestions have greatly enriched my work and provided valuable guidance for the research I have been pursuing.

The Department of Statistics at the University of Waterloo has offered countless opportunities and brought together many inspiring people who have guided my statistical journey over the past four years. I am deeply grateful to them for taking a chance on me and accepting me into the PhD program.

Lastly, my family and friends deserve far more than a simple expression of thanks for their unwavering support and encouragement throughout this journey. Their constant presence, patience, and steadfast belief in me have been a continuous source of strength and motivation, sustaining me through both challenges and successes. All my achievements would not have been possible without their love and confidence in my abilities, for which I remain profoundly grateful.

Table of Contents

Examining Committee Membership	ii
Author’s Declaration	iii
Statement of Contributions	iv
Abstract	v
Acknowledgements	vi
List of Figures	xi
List of Tables	xviii
1 Introduction	1
1.1 Literature Review	2
1.1.1 ODE Fidelity	3
1.1.2 Sparse Regularization	5
1.1.3 Model Inadequacy	6
1.2 Outline of the Thesis	7

2	A Bayesian Collocation Integral Method for Parameter Estimation in Ordinary Differential Equations	8
2.1	Introduction	8
2.2	Methodology	10
2.2.1	Bayesian Structure	10
2.2.2	Integral Estimation	12
2.2.3	The Choice of Smoothing Parameter	13
2.2.4	Practical Implementation	16
2.3	Simulation Study	16
2.3.1	Benchmark System and Setup	16
2.3.2	Performance Assessment	19
2.3.3	Results	20
2.4	Real Data Analysis	25
2.5	Discussion	26
3	A Bayesian Collocation Integral Method for System Identification of Ordinary Differential Equations	28
3.1	Introduction	28
3.2	Methodology	31
3.2.1	Bayesian Structure	31
3.2.2	Integral Estimation	34
3.2.3	Identifiability	35
3.2.4	Practical Implementation	35
3.3	Simulation Study	36
3.3.1	Benchmark System and Setup	36
3.3.2	Performance Assessment	38
3.3.3	Results	40
3.4	Real Data Analysis	43
3.5	Conclusion and Discussion	45

4	A Bayesian Collocation Integral Method for Model Validation of Ordinary Differential Equations	50
4.1	Introduction	50
4.2	Methodology	52
4.2.1	Bayesian Structure	52
4.2.2	Inadequacy Estimation and Model Validation	53
4.2.3	Model Selection	54
4.2.4	Practical Implementation	56
4.3	Simulation	56
4.3.1	Benchmark System and Setup	56
4.3.2	Performance Assessment	58
4.3.3	Results	59
4.4	Real Data Analysis	62
4.5	Discussion	67
5	Discussion and Future Work	68
	References	70
	APPENDICES	78
A	Implementation Details	79
A.1	Details of sampling algorithm	79
A.2	Software Setup	80
B	Additional Results for Chapter 2	81
B.1	Simulation studies: additional results	81
B.1.1	FitzHugh-Nagumo (FN) system	81
B.1.2	Protein transduction system	90
B.1.3	Order of basis functions	97

B.1.4	Consistency of selecting λ	100
B.1.5	Irregular Observations	105
B.2	Real data analysis: additional results	109
C	Additional Results for Chapter 3	114
C.1	Choice of quadrature points	114
C.2	Simulation studies: additional results	114
C.2.1	Additive ODEs	114
C.2.2	Absolute value functions	121
C.2.3	Lotka-Volterra equations	123
C.3	Real data analysis: additional results	126
D	Additional Results for Chapter 4	130
D.1	Simulation studies: additional results	130
D.1.1	Sensitivity of τ	130

List of Figures

2.1	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 41 observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	18
2.2	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 21 observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	22
2.3	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 11 observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	23
2.4	Estimated trajectories for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	24
3.1	The true trajectories $\mathbf{x}_1, \dots, \mathbf{x}_6$ of the simulated ODE system (3.15) in the time interval $[0, 20]$	36
3.2	The pointwise coverage probabilities of each trajectory in the simulated ODE system for 41 observations over 100 simulation runs when SNR=10 (black line). The red horizontal line indicates the nominal level 95%.	41

3.3	Estimated trajectories (gene expression levels) for gene 45 (FYB), 46 (IL-2R γ), 49 (apoptosis) and 55 (IL3R α). The black triangles represent the real data and the red solid lines are the estimated trajectories. The red shaded areas are central 95% pointwise credible intervals for the estimated trajectories.	47
3.4	Graph of T-cell activation GRN formed by 58 genes. Each gene is represented as a node, and arrows represent the direction of influence. FYB (gene 45) is marked in pink, with its connections highlighted in red.	48
3.5	The estimated regulation functions of genes influenced by FYB (gene 45). In each panel, the x-axis represents the estimated expression level of FYB, and the y-axis measures the estimated regulation effect, which varies with the estimated expression level of FYB.	49
4.1	Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with low-noise settings. The black dashed line represents zero line. The red solid line represents the average estimated inadequacy function, and the red shaded area represents the pointwise central 95% intervals of all estimated inadequacy functions.	58
4.2	Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with high-noise settings. The black dashed line represents zero line. The red solid line represents the average estimated inadequacy function, and the red shaded area represents the pointwise central 95% intervals of all estimated inadequacy functions.	59
4.3	Estimated trajectories for x_1 (left panel) and x_2 (right panel) in Model 1 (4.7), over 100 simulation runs with low-noise settings. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	61
4.4	Estimated trajectories for x_1 (left panel) and x_2 (right panel) in Model 1 (4.7), over 100 simulation runs with high-noise settings. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	61

4.5	Estimated inadequacy functions for the population of <i>Paramecium aurelia</i> (x_1) and <i>Paramecium caudatum</i> (x_2) in a 19-day period for LV competition model. The black dashed line represent the zero line and the red solid line is the estimated inadequacy function. The red shaded area is a central 95% pointwise credible interval for the estimated inadequacy function.	64
4.6	Estimated inadequacy functions for the population of <i>Paramecium aurelia</i> (x_1) and <i>Paramecium caudatum</i> (x_2) in a 19-day period for four LV predator-prey type models. The black dashed line represent the zero line and the red solid line is the estimated inadequacy function. The red shaded area is a central 95% pointwise credible interval for the estimated inadequacy function.	65
4.7	Estimated trajectories for the population of <i>Paramecium aurelia</i> (x_1) and <i>Paramecium caudatum</i> (x_2) in a 19-day period for five LV type models. The black triangles represent the real data. The blue, light blue, orange, green and red solid lines are the estimated trajectories for Model 1, 2, 3, 4 and 5, respectively.	66
B.1	Estimated parameter RMSEs for a, b and c respectively in the FN system, over 100 simulation runs with 41 observations.	82
B.2	Estimated parameter RMSEs for a, b and c respectively in the FN system, over 100 simulation runs with 21 observations.	83
B.3	Estimated parameter RMSEs for a, b and c respectively in the FN system, over 100 simulation runs with 11 observations.	83
B.4	Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 41 observations.	84
B.5	Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 21 observations.	84
B.6	Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 11 observations.	85
B.7	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations of one random simulation run. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	87

B.8	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 21 observations of one random simulation run. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	88
B.9	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 11 observations of one random simulation run. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	89
B.10	Estimated trajectory RMSEs for each component in the protein transduction system, over 100 simulation runs with low noise.	93
B.11	Estimated trajectories for each component in the protein transduction system, over 100 simulation runs with low-noise setting. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	94
B.12	Estimated trajectory RMSEs for each component in the protein transduction system, over 100 simulation runs with high noise.	95
B.13	Estimated trajectories for each component in the protein transduction system, over 100 simulation runs with high-noise setting. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	96
B.14	Estimated parameter RMSEs in the FN system using the proposed method, over 100 simulation runs with 41 observations.	98
B.15	Estimated trajectory RMSEs for each component in the FN system using the proposed method, over 100 simulation runs with 41 observations. . . .	99
B.16	Selected λ in the FN system using the proposed method, over 100 simulation runs with 41 observations.	100
B.17	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations with $\lambda = 10^3$. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	102

B.18	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations with $\lambda = 5 \times 10^3$. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	103
B.19	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations with $\lambda = 10^4$. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	104
B.20	True trajectories for V (top panel) and R (bottom panel) in the FN system and the spacing of the 11 irregular observations. The black line represents the true trajectories and the red points are the locations of the observed time points.	105
B.21	Estimated parameter RMSEs for a, b and c respectively in the FN system, over 100 simulation runs with 11 irregular observations.	106
B.22	Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 11 irregular observations.	107
B.23	Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 11 irregular observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.	108
B.24	Estimated trajectories from the derivative method for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	110
B.25	Estimated trajectories from MAGI for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	111

B.26	Estimated trajectories from FGPGM for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.	112
C.1	The pointwise coverage probabilities of each trajectory in the simulated ODE system for 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.	115
C.2	The pointwise coverage probabilities of each trajectory in the simulated ODE system for 41 observations over 100 simulation runs when SNR=4 (black line). The red horizontal line indicates the nominal level 95%. . . .	116
C.3	The pointwise coverage probabilities of each trajectory in the simulated ODE system for 21 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.	118
C.4	The pointwise coverage probabilities of each trajectory in the simulated ODE system for 21 observations over 100 simulation runs when SNR=10 (black line). The red horizontal line indicates the nominal level 95%.	118
C.5	The pointwise coverage probabilities of each trajectory in the simulated ODE system for 21 observations over 100 simulation runs when SNR=4 (black line). The red horizontal line indicates the nominal level 95%. . . .	119
C.6	The pointwise coverage probabilities of each trajectory in the simulated ODE system with $k = 8$ for 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.	120
C.7	The true trajectories $\mathbf{x}_1, \dots, \mathbf{x}_6$ of the simulated ODE system (C.1) in the time interval $[0, 20]$	121
C.8	The pointwise coverage probabilities of each trajectory in the simulated ODE system (C.1) for 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%. . . .	123
C.9	The pointwise coverage probabilities of each trajectory in the simulated ODE system Lotka-Volterra equations with 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.	125

D.1	Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with low-noise settings. The black dashed line represents zero line. The red, blue and green solid line represents the average estimated inadequacy function for τ , 2τ , and 5τ , respectively. The red, blue and green shaded area represents the corresponding pointwise central 95% intervals of all estimated inadequacy functions for τ , 2τ , and 5τ , respectively.	131
D.2	Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with low-noise settings. The black dashed line represents zero line. The red, blue and green solid line represents the average estimated inadequacy function for τ , 0.5τ , and 0.2τ , respectively. The red, blue and green shaded area represents the corresponding pointwise central 95% intervals of all estimated inadequacy functions for τ , 0.5τ , and 0.2τ , respectively.	132

List of Tables

2.1	Mean of the parameter estimates with RMSE in parentheses for the FN system across 100 simulated runs. ¹	20
2.2	Median of trajectory RMSEs (with IQR in parentheses) for each component of the FN system across the 100 simulated runs. ²	21
2.3	Parameter estimates with 95% credible interval in parentheses, when using the LV model (2.10) to fit the hare and lynx data for 1908-1928. Here x_{01} and x_{02} denote the estimated initial conditions for x_1 and x_2 , respectively.	26
3.1	Comparison of methods for 41 observations in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.	40
4.1	Mean of the parameter estimates with RMSE in parentheses for Model 1 (4.7) across 100 simulated runs.	60
4.2	Median of trajectory RMSEs (with IQR in parentheses) for each component of Model 1 (4.7) and Model 2 (4.8) across 100 simulated runs.	60
4.3	Parameter estimates results using five LV type models to fit the population of Paramecium aurelia (x_1) and Paramecium caudatum (x_2) in a 19-day period.	66
B.1	Average of the norms of each estimated component in the FN system across 100 simulation runs.	86
B.2	Average running time (minutes) of each method for the FN system across 100 simulation runs.	90
B.3	Median of trajectory RMSEs (with IQR in parentheses) for each component of the protein transduction system across 100 simulation runs.	92

B.4	Average norm of each estimated component of the protein transduction system across the 100 simulation runs.	97
B.5	Average running time (minutes) of each method for the protein transduction system across 100 simulation runs.	97
B.6	Mean of the parameter estimates with RMSE in parentheses for the FN system with 41 observations using the proposed method across the 100 simulation runs.	98
B.7	Median of trajectory RMSEs (with IQR in parentheses) for each component of the FN system with 41 observations using the proposed method across 100 simulation runs.	99
B.8	Average norm for each estimated component of the FN system with 41 observations using the proposed method across 100 simulation runs.	100
B.9	Average running time (minutes) of the FN system with 41 observations using the proposed method across 100 simulation runs.	100
B.10	Median of trajectory RMSEs between $\lambda = 10^3$ and $\lambda = 5 \times 10^3/10^4$ (with IQR in parentheses) for each component of the FN system with 41 observations across the 100 simulated runs.	101
B.11	Average norm for each estimated component of the FN system with 41 observations using the proposed method across 100 simulation runs.	101
B.12	Mean of the parameter estimates with RMSE in parentheses for the FN system with 11 irregular observations across the 100 simulation runs.	106
B.13	Median of trajectory RMSEs (with IQR in parentheses) for each component of the FN system with 11 irregular observations across 100 simulation runs.	107
B.14	Average of the norms of each estimated component in the FN system with 11 irregular observations across 100 simulation runs.	107
B.15	Parameter estimates with 95% credible interval in parentheses, when using the LV model to fit the hare and lynx data for 1908-1928. Here x_{01} and x_{02} denote the estimated initial conditions for x_1 and x_2 , respectively.	109
B.16	Running time (minutes) of each method when using the LV model to fit the hare and lynx data for 1908-1928.	112
B.17	RMSEs between the reconstructed trajectories and the observations using the LV model and the hare and lynx data for 1908-1928.	113

C.1	Comparison of methods for 21 observations in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.	117
C.2	Comparison of number of equations for 41 observations with $\text{SNR} = 25$ in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.	120
C.3	Comparison of methods for system (C.1) with 41 observations and $\text{SNR} = 25$ in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.	122
C.4	Comparison of methods for Lotka-Volterra equations with 41 observations and $\text{SNR} = 25$ in trajectory recovery, ODE vanishing components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.	125
C.5	Variable selection results of T-cell activation gene regulation network. . .	127
C.6	Frequency of top 15 genes with highest number of outward influence genes over 50 independent trials in T-cell activation gene regulation network with different η values.	128
C.7	Frequency of each gene being an outward influence gene of FYB (gene 45) over 50 independent trials in T-cell activation gene regulation network with different η	129
D.1	Mean of the parameter estimates with RMSE in parentheses for Model 1 (4.7) under different τ across 100 simulated runs.	131

Chapter 1

Introduction

Ordinary differential equations (ODEs) are widely used in many scientific areas, including physics, ecology and biomedical sciences, to model the behavior of complex dynamic systems. We consider the setup with a system of ODEs taking the form

$$\mathbf{x}'(t) \equiv \begin{bmatrix} \frac{dx_1(t)}{dt} \\ \vdots \\ \frac{dx_I(t)}{dt} \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}(t), t) \\ \vdots \\ f_I(\mathbf{x}(t), t) \end{bmatrix} \equiv \mathbf{f}(\mathbf{x}(t), t), \quad t \in [t_1 = 0, t_J], \quad (1.1)$$

where the vector $\mathbf{x}(t) = (x_1(t), \dots, x_I(t))^\top$ denotes the set of I variables that evolve over time t , $\mathbf{f}$ can be a known or unknown function that specifies the form of the system derivatives. We denote the initial condition by $\mathbf{x}_0 \equiv \mathbf{x}(0)$. In time-course experiments, the system is often observed only at a sparse set of time points $\{t_1, \dots, t_J\}$ and subject to measurement error, so that at time t_j we have the noisy observation $\mathbf{y}_j \in \mathbb{R}^I$ according to

$$\mathbf{y}_j = \mathbf{x}(t_j) + \boldsymbol{\epsilon}_j, \quad j = 1, \dots, J, \quad (1.2)$$

where $\boldsymbol{\epsilon}_j \in \mathbb{R}^I$ is an independent error term.

When the functional form of $\mathbf{f}$ is known, we specifically consider the setup that writes the system of ODEs (1.1) as

$$\mathbf{x}'(t) = \begin{bmatrix} f_1(\mathbf{x}(t), \boldsymbol{\theta}, t) \\ \vdots \\ f_I(\mathbf{x}(t), \boldsymbol{\theta}, t) \end{bmatrix} \equiv \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}, t), \quad t \in [t_1 = 0, t_J], \quad (1.3)$$

where $\boldsymbol{\theta}$ is the unknown time-independent parameter vector. Therefore, the primary goal can be to infer the parameter vector $\boldsymbol{\theta}$ based on the observed data $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_J)$. For nonlinear ODEs, where analytical solutions to (1.3) are generally unavailable, various approaches for estimating $\boldsymbol{\theta}$ have been developed. These approaches can be broadly classified into three types based on how they solve or approximate $\mathbf{x}(t)$. The first relies on numerical solvers to explicitly solve the ODE, updating parameters iteratively to find the optimal fit. However, repeated use of such solvers can be computationally intensive (Ramsay et al., 2007; Qi and Zhao, 2010). Alternatively, collocation methods (Ramsay et al., 2007; Wang et al., 2022) and methods based on Gaussian processes (GP) (Calderhead et al., 2008; Dondelinger et al., 2013; Wenk et al., 2019; Yang et al., 2021; Tronarp et al., 2022; Wu and Lysy, 2024) that bypass the numerical integration can alleviate this difficulty.

In high-dimensional time-course data, where the functional relationships among variables are often uncertain, the functional form $\mathbf{f}$ is typically assumed to be unknown to allow for greater modeling flexibility. Consequently, identifying the structure of the underlying ODE system becomes a key objective. This requires suitable model assumptions for (1.1), such as linear ODEs (Lu et al., 2011); parametric ODEs with two-way interactions (Zhang et al., 2015); sparse additive ODEs (Wu et al., 2014; Chen et al., 2017); and kernel ODEs (Dai and Li, 2022). To simultaneously perform variable selection and parameter estimation, sparsity-inducing penalties have been widely studied, including the smoothly clipped absolute deviation (Lu et al., 2011) and the adaptive group lasso (Wu et al., 2014; Chen et al., 2017; Zhang et al., 2022).

Regardless of the assumptions made on the ODE model setup, most existing methods entail estimating the derivatives of ODE solutions from noisy observations. Since the performance depends heavily on a reliable estimate of the derivatives, it can be inefficient and challenging (Chen et al., 2017). To address this issue, Dattner and Klaassen (2015), Chen et al. (2017) and Dai and Li (2022) considered an integrated form of the ODEs (1.1) from the frequentist perspective. Nevertheless, to our knowledge, this integral reformulation has not been extensively developed within a Bayesian framework. To address this gap, we consider a Bayesian integral-based method for both known and unknown function form $\mathbf{f}$, which can better quantify the parameter uncertainty compared to frequentist approaches (Wang et al., 2022).

1.1 Literature Review

Let y_{ij} denote the noisy observation of $x_i(t)$ made at time t_j for $i = 1, \dots, I$ and $j = 1, \dots, J$, i.e., the i th component in $\mathbf{y}_j$. With the ODE structure (1.1) and the measurement model

(1.2) aforementioned, for concreteness in the following exposition we treat the measurement error ϵ_{ij} as independent Gaussian with variance σ_i^2 , i.e., $\epsilon_{ij} \sim N(0, \sigma_i^2)$. This gives the ideal likelihood function of the observations:

$$p(\mathbf{y}|\mathbf{x}(t), \boldsymbol{\sigma}) \propto \left(\prod_{i=1}^I \prod_{j=1}^J \sigma_i^2 \right)^{-1/2} \exp \left\{ - \sum_{i=1}^I \left[\sum_{j=1}^J \frac{(y_{ij} - x_i(t_j))^2}{2\sigma_i^2} \right] \right\}. \quad (1.4)$$

1.1.1 ODE Fidelity

If given the function form of $\mathbf{f}$ (1.3), the numerical solvers can be used to obtain the values of $x_i(t)$ (as a function of $\boldsymbol{\theta}$ and $\mathbf{x}_0$) for this likelihood calculation. However, this can be computationally intensive; in addition, this form can not be generalized to the case for an unknown function form $\mathbf{f}$ (1.1). Pioneered by Varah (1982), $x_i(t)$ can be approximated using an expansion of L cubic spline basis functions $\Phi(t) = (\phi_1(t), \phi_2(t), \dots, \phi_L(t))^\top$, i.e.,

$$\hat{x}_i(t) = \Phi(t)^\top \mathbf{c}_i, \quad (1.5)$$

where the column vector $\mathbf{c}_i$ denotes the basis coefficients for component $i = 1, \dots, I$. Note that under this approximation, an estimate of the initial condition of each component is given by $\hat{x}_i(0) = \Phi(0)^\top \mathbf{c}_i$. As a result, $x_i(t)$ does not need an explicit estimation. Alternatively, from the Bayesian perspective, a GP prior can be imposed on $x_i(t)$ (Calderhead et al., 2008):

$$\hat{x}_i(t) \sim GP(\mu_i, \mathcal{K}_i), \quad (1.6)$$

where $\mathcal{K}_i : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ denotes a positive definite covariance kernel for the GP and $\mu_i : \mathbb{R} \rightarrow \mathbb{R}$ is the mean function. From a related perspective, Tronarp et al. (2022) and Wu and Lysy (2024) employed probabilistic approaches that impose a Gaussian Markov process prior on $\mathbf{x}(t)$,

$$\hat{\mathbf{x}}(t_{j+1}) \mid \hat{\mathbf{x}}(t_j) \sim \mathcal{N}(\mathbf{Q}(\Delta(t))\hat{\mathbf{x}}(t_j), \mathbf{R}(\Delta(t))), \text{ for } j = 1, \dots, J, \quad (1.7)$$

where $\Delta(t) = t_{j+1} - t_j$ is usually evenly spaced, and $\mathbf{Q}(\Delta(t))$ and $\mathbf{R}(\Delta(t))$ are determined by the Gaussian Markov process prior on $\mathbf{x}(t)$. Regardless of which methods bypass the numerical integration, incorporating the information about the parameters via ODE is a vital part of the model. Therefore, numerous methods consider quantifying the discrepancy between $\hat{\mathbf{x}}'(t)$ and the ODE system $\mathbf{f}$ (Varah, 1982; Ramsay et al., 2007; Wang et al., 2022; Wu et al., 2014; Zhang et al., 2022; Yang et al., 2021; Tronarp et al., 2022; Wu and Lysy,

2024), namely $\|\hat{\mathbf{x}}'(t) - \mathbf{f}(\hat{\mathbf{x}}(t), t)\|$. For example, with the given form of $\mathbf{f}$, Varah (1982) considered a two-step collocation method. They first fitted the spline estimates (1.5) to the observations, and then used these estimates to solve the optimization problem,

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \int_{t_1}^{t_J} \|\hat{\mathbf{x}}'(t) - f(\hat{\mathbf{x}}(t), \boldsymbol{\theta}, t)\|_2^2 dt. \quad (1.8)$$

On the other hand, the manifold-constrained Gaussian process inference (MAGI, Yang et al., 2021) constrains the GP (1.6) to satisfy the ODE model (1.3) by defining a random variable W as

$$W = \sup_{t \in [t_1, t_J], i \in \{1, \dots, I\}} |\hat{x}'_i(t) - f_i(\hat{\mathbf{x}}(t), \boldsymbol{\theta}, t)|, \quad (1.9)$$

and considers a posterior distribution for $\mathbf{x}(t)$ and $\boldsymbol{\theta}$ given the observations $\mathbf{y}$ and constraint $W \equiv 0$. With this principled Bayesian construction, the MAGI provides promising estimation accuracy and computational efficiency.

However, Dattner and Klaassen (2015) pointed out that the direct incorporation of ODE system (1.1) involves the estimation of derivative $\mathbf{x}'(t)$, which may be rather inaccurate given the noisy and sparse data. In fact, by Theorem 1 of Stone (1985), the convergence rate of the spline-based derivative estimator is slower than that of the function estimator. Therefore, if regression splines are employed to estimate derivatives from noisy and sparse observations, the results might not be reliable. Similarly, for GP priors, the derivative estimator of the posterior distribution has a slower convergence rate for sparser observations, as shown in Liu and Li (2022). As a result, Dattner and Klaassen (2015) proposed an improvement to (1.8) under the linear ODE model assumption,

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \int_{t_1}^{t_J} \left\| \hat{\mathbf{x}}(t) - \left\{ \int_0^t f(\hat{\mathbf{x}}(s), s) ds \right\} - \hat{\mathbf{x}}(0) \right\|_2^2 dt, \quad (1.10)$$

where $\hat{\mathbf{x}}(t)$ is the spline approximation (1.5). This integral modification to (1.1) can achieve better performance than the direct incorporation theoretically and empirically, but it is not feasible for the GP-based approach (1.6). Consequently, we consider the spline approximation (1.5) with this integral modification technique for both known and unknown function forms of $\mathbf{f}$.

1.1.2 Sparse Regularization

If $\mathbf{f}$ in (1.1) has an unknown form, appropriate assumptions must be imposed on the ODE system to identify its structure. For example, [Lu et al. \(2011\)](#) assumed that (1.1) follows a linear form, while [Zhang et al. \(2015\)](#) relaxed this parametric assumption by allowing two-way interactions. However, both approaches can be overly restrictive in practical applications, particularly in contexts such as gene regulatory networks ([Wu et al., 2014](#)). To further relax the assumptions in (1.1), [Wu et al. \(2014\)](#) proposed sparse additive ODEs, where the additive components are approximated using basis function expansions,

$$x'_i(t) = a_{i0} + \sum_{p=1}^I f_{ip}(x_p(t)) = a_{i0} + \sum_{p=1}^I \Psi(x_p(t))^\top \mathbf{a}_{ip}, \quad i = 1, 2, \dots, I, \quad (1.11)$$

where a_{i0} is the intercept term, $\Psi = (\psi_1, \psi_2, \dots, \psi_D)^\top$ is the basis functions and the column vector $\mathbf{a}_{ip} \in \mathbb{R}^D$ denotes the corresponding basis coefficients for component $i, p = 1, \dots, I$. Accordingly, the problem of interest can be treated as estimating and selecting the non-zero element f_{ip} or $\mathbf{a}_{ip}$, and a variety of sparse penalization methods have been considered. From a frequentist perspective, [Chen et al. \(2017\)](#) imposed a sparse penalty on the basis expansion of f_{ip} ,

$$\text{pen}_{\lambda_i} = \lambda_i \sum_{p=1}^I \left[\frac{1}{J} \sum_{s=t_1}^{t_J} \left\{ \int_0^s \Psi(x_p(s))^\top \mathbf{a}_{ip} \, ds \right\}^2 \right]^{1/2}, \quad p = 1, \dots, I, \quad (1.12)$$

where $\lambda_i > 0$ denotes a sparsity-inducing tuning parameter. On the other hand, [Wu et al. \(2014\)](#) and [Zhang et al. \(2022\)](#) applied the sparse penalty to $\mathbf{a}_{ip}$,

$$\text{pen}_{\lambda_i} = \lambda_i \sum_{p=1}^I w_{ip} \|\mathbf{a}_{ip}\|_2, \quad p = 1, \dots, I, \quad (1.13)$$

where the weight $w_{ip} = \|\mathbf{a}_{ip}\|_2^{-v}, v > 0$ if $\|\mathbf{a}_{ip}\|_2 > 0$ and $w_{ip} = \infty$ otherwise. Both pen_{λ_i} (1.12) and (1.13) are adaptive group lasso penalties, which can provide estimation efficiency and selection consistency simultaneously ([Wang and Leng, 2008](#)). In the Bayesian framework, similar behavior can be achieved by placing spike-and-slab group lasso priors on each $\mathbf{a}_{ip}$ ([Bai et al., 2022](#)),

$$\tilde{\pi}_0(\mathbf{a}_{ip}|\theta_{ip}) = \theta_{ip}\delta(\mathbf{a}_{ip}|\eta_0) + (1 - \theta_{ip})\delta(\mathbf{a}_{ip}|\eta_1), \quad \text{with } \eta_1 \gg \eta_0 > 0, \quad (1.14)$$

where $\theta_{ip} \in (0, 1)$ is the mixing proportion, and $\delta(\mathbf{a}_{ip}|\eta) \propto \eta^D \exp(-\eta\|\mathbf{a}_{ip}\|_2)$ denotes the group lasso density. The parameter η_0 corresponds to the spike component, which strongly shrinks $\mathbf{a}_{ip}$ toward zero, while η_1 represents the slab component that allows for nonzero values. Thus, following [Wu et al. \(2014\)](#) and [Zhang et al. \(2022\)](#), we consider additive ODEs with the basis expansions to their components (1.11). Moreover, a spike-and-slab group lasso penalty ([Bai et al., 2022](#)) is imposed in our Bayesian model.

1.1.3 Model Inadequacy

Even when the functional form for $\mathbf{f}$ is known in (1.3), there may be several competing models that produce similar dynamics. However, due to the limited flexibility of mathematical models or the limited understanding of system dynamics, ODEs may not capture the full variability observed in real data, leading to discrepancies between the model outputs $\mathbf{x}(t)$ and the observed data $\mathbf{y}$ ([Kim et al., 2013](#)). Although measurement error from noisy observations can cause discrepancies since ODE trajectories are inherently deterministic, they can also arise from model inadequacies ([Kennedy and O'Hagan, 2001](#)). To systematically capture discrepancies caused by model inadequacies and validate the ODE models, [Kennedy and O'Hagan \(2001\)](#) formulated model inadequacy as a function and introduced it to the measurement model (1.2),

$$\mathbf{y}_j = \mathbf{x}(t_j) + \boldsymbol{\epsilon}_j + \mathbf{d}(t_j), \quad j = 1, \dots, J, \quad (1.15)$$

where $\mathbf{d}(t_j) = (d_1(t_j), \dots, d_I(t_j))^\top$ denotes the unspecified function representing the potential model inadequacy of $\mathbf{y}_j$ at time point j . [Bayarri et al. \(2007\)](#) pointed out that if the inadequacy function $\mathbf{d}(t_j)$ and its uncertainty can be estimated from the data, the validity of the model can be systematically assessed. However, since inadequacy function and model parameters are often highly confounded, they must be jointly estimated. Within a Bayesian framework, [Kim et al. \(2013\)](#) imposed a multivariate normal prior on $\mathbf{d}_i$,

$$\mathbf{d}_i \sim \text{MVN}\left(\mathbf{0}, \Delta = [\exp\{-(t_j - t_k)^2\}]_{jk}\right), \quad j, k = 1, \dots, J. \quad (1.16)$$

However, their method lacks flexibility in specifying the measurement error distribution. On the other hand, from a frequentist perspective, [Kim \(2016\)](#) and [Kim and Kim \(2019\)](#) modeled $\mathbf{d}(t_j)$ as a nonlinear smoothing function of time t_j , conditional on a given parameter vector $\boldsymbol{\theta}$. Based on the measurement model (1.15), we synthesize these ideas by modeling the inadequacy function using a basis function expansion, such as cubic B-splines, for validating general nonlinear ODE models.

1.2 Outline of the Thesis

In this thesis, we develop Bayesian integral-based methods for both known function and unknown function forms $\mathbf{f}$ in ODE models (1.1). In Chapter 2, we propose a Bayesian collocation integral method to infer the parameter $\boldsymbol{\theta}$ in (1.3). This chapter is adapted from the paper “A Bayesian Collocation Integral Method for Parameter Estimation in Ordinary Differential Equations” that was published in the *Journal of Computational and Graphical Statistics* (Xu et al., 2024). In Chapter 3, we propose a Bayesian integral-based collocation model with spike-and-slab group lasso penalty to recover the ODE structure $\mathbf{f}$ in (1.1). This chapter is adapted from the paper “A Bayesian Collocation Integral Method for System Identification in Ordinary Differential Equations”, which has been submitted for publication and is currently under revision. In Chapter 4, we propose a Bayesian collocation method based on Chapter 2 and measurement model (1.15) to validate general nonlinear ODE models. Finally, we conclude our thesis and discuss some potential future work in Chapter 5.

Chapter 2

A Bayesian Collocation Integral Method for Parameter Estimation in Ordinary Differential Equations

2.1 Introduction

Assume the function $\mathbf{f}$ in (1.3) takes the known form, then the problem of interest is to infer $\boldsymbol{\theta}$ given the observed data $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_J)$. For nonlinear ODEs, analytical solutions to (1.3) are typically unavailable; a numerical method such as the Runge–Kutta algorithm is required to solve the ODEs. To account for the measurement model (1.2), a numerical solver could be combined with nonlinear least squares (NLS) to estimate $\boldsymbol{\theta}$, for example, by minimizing $\sum_{j=1}^J \|\mathbf{y}_j - \mathbf{x}(t_j)\|^2$. However, this type of approach involves many iterative updates to the parameters and initial conditions, and the repeated use of the numerical solver can be computationally intensive, especially for stiff systems or for discontinuous inputs (Ramsay et al., 2007; Qi and Zhao, 2010).

As an alternative, collocation methods can alleviate this difficulty: a basis function expansion is used to approximate the ODE solution $\mathbf{x}(t)$, i.e., by letting $\hat{\mathbf{x}}(t) = \mathbf{c}^\top \Phi(t)$, where $\Phi(t)$ is a vector of basis functions and $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_I)$ with each $\mathbf{c}_i$ representing the vector of basis coefficients for $x_i(t)$. Then the derivative $\hat{\mathbf{x}}'(t)$ also has a basis function expression, so that its discrepancy from the actual ODE model, namely $\|\hat{\mathbf{x}}'(t) - \mathbf{f}(\hat{\mathbf{x}}(t), \boldsymbol{\theta}, t)\|$, has a convenient analytical form. Thus, the ODEs do not need to be solved explicitly for the inference procedure. Pioneering this approach from the frequentist perspective, Varah

(1982) proposed a two-step procedure: the first step fits the spline estimates to the observations, and the second step estimates the ODE parameters via least squares. Ramsay et al. (2007) noted that this two-step procedure may only work well when a satisfactory estimate of $\mathbf{x}'(t)$ is obtained in the smoothing step; to circumvent this limitation, they proposed a penalized spline method along with profiled estimation techniques to fit the observations and ODEs together. On the other hand, Bayesian approaches may better quantify parameter uncertainty and hence Bayesian hierarchical collocation models have also been developed (Wang et al., 2022); their method is attractive for simple equations, but lacks general guidelines for selecting the hyperparameters for $\boldsymbol{\theta}$ and the smoothing parameter.

Another type of Bayesian approaches for ODE parameter inference involves the use of Gaussian processes (GPs). By imposing a GP prior on $\mathbf{x}(t)$, the joint distribution of $\mathbf{x}'(t)$ and $\mathbf{x}(t)$ at any finite set of time points is multivariate normal. Thus, a GP approximation to the ODE system can also potentially bypass the need for numerical integration. The idea of GP-based gradient matching was first explored in the work of Calderhead et al. (2008) and Dondelinger et al. (2013). A subsequent refinement to the structure of the probabilistic model, named fast Gaussian process-based gradient matching (FGPGM, Wenk et al., 2019), gave improved parameter estimation results. These GP-based methods encounter a conceptual incompatibility between the stochastic nature of the GP and the deterministic nature of the actual ODE system $\mathbf{f}$. Yang et al. (2021) proposed the manifold-constrained Gaussian process inference (MAGI) to address this problem. The MAGI method explicitly incorporates the ODE system into the GP as a manifold constraint via conditional probability; this principled Bayesian construction provides promising estimation accuracy and computational efficiency in a variety of examples.

However, these GP and collocation methods that bypass numerical integration entail estimating the derivatives of ODE solutions from noisy observations. As pointed out by Chen et al. (2017), this can be inefficient and challenging. This issue becomes even more severe when only sparse time-course observations are available. Starting from the two-step collocation procedure, Dattner and Klaassen (2015) proposed an improvement under the assumption that $\mathbf{f}$ is a linear function of $\boldsymbol{\theta}$. The method integrates both sides of (1.3) and estimates the integral $\int \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}, t) dt$ rather than estimating the derivative $\mathbf{x}'(t)$. Following this technique, Chen et al. (2017) proposed an integral-based method to consistently recover the true network structure, especially in high dimensions. Nevertheless, this method needs to assume that each dynamic system is specified as a system of additive ODEs. Dattner and Klaassen (2015) and Chen et al. (2017) both demonstrated that integral-based methods have better performance than derivative-based methods theoretically and empirically.

In this paper, we propose to improve previous approaches by introducing integral esti-

mation to the Bayesian hierarchical collocation model. Specifically, we approximate $\mathbf{x}(t)$ using the expansion of basis functions such as cubic B-splines to avoid the expensive use of numerical solvers (Ramsay et al., 2007). This choice allows us to integrate the B-spline approximation $\hat{\mathbf{x}}(t)$, which is not feasible for a GP-based approach. A smoothing parameter is combined with the integrated ODE constraints to control the trade-off between the fit to data and the fidelity to the ODEs. Inspired by Ramsay et al. (2007) and Qi and Zhao (2010), we design an algorithm that automatically selects the smoothing parameter. The specification of priors for the parameters is flexible: the method works well with generic priors, and can also accommodate custom prior specifications. The choice of measurement error distribution is also flexible. To draw samples from the posterior distribution, we use the no-U-turn sampler (NUTS, Homan and Gelman, 2014), which can be more efficient than traditional Markov chain Monte Carlo (MCMC) samplers. Our method is designed to handle general nonlinear ODE systems. We demonstrate the accuracy of our method for estimating the parameters and recovering the system trajectories in the simulation studies, via comparisons with other Bayesian methods. A real data example is also provided.

The remainder of the paper is organized as follows. Section 2.2 provides the formulation of our Bayesian method, including a detailed description of the algorithm and computational techniques. Illustrations of the proposed method are presented in Section 2.3 for simulated data and in Section 2.4 for real data. Finally, Section 2.5 concludes the paper.

2.2 Methodology

2.2.1 Bayesian Structure

Using numerical solvers to obtain the values of $x_i(t)$ (as a function of $\boldsymbol{\theta}$ and $\mathbf{x}_0$) needed for this likelihood calculation can be computationally intensive. Thus, following Ramsay et al. (2007), we approximate $x_i(t)$ using an expansion of L cubic B-spline basis functions $\Phi(t) = (\phi_1(t), \phi_2(t), \dots, \phi_L(t))^\top$, i.e.,

$$\hat{x}_i(t) = \Phi(t)^\top \mathbf{c}_i, \quad (2.1)$$

where the column vector $\mathbf{c}_i$ denotes the basis coefficients for component $i = 1, \dots, I$. Note that under this approximation, an estimate of the initial condition of each component is given by $\hat{x}_i(0) = \Phi(0)^\top \mathbf{c}_i$; namely, no explicit estimation of $\mathbf{x}_0$ is needed. In practice, the performance of our method is not sensitive to the choice of the order of the basis functions. However, for the sake of computational efficiency, we recommend cubic B-spline bases, i.e.,

B-spline of order four, in our numerical studies; see Section B.1.3 of the appendix for more details.

Consequently, the likelihood function (1.4) is replaced by the spline-approximated version

$$p(\mathbf{y}|\mathbf{c}, \boldsymbol{\sigma}) \propto \left(\prod_{i=1}^I \prod_{j=1}^J \sigma_i^2 \right)^{-1/2} \exp \left\{ - \sum_{i=1}^I \left[\sum_{j=1}^J \frac{(y_{ij} - \Phi(t_{ij})^\top \mathbf{c}_i)^2}{2\sigma_i^2} \right] \right\}. \quad (2.2)$$

We need to assign an appropriate prior to facilitate the estimation of the basis coefficients $\mathbf{c}_i$, which can incorporate information about $\boldsymbol{\theta}$ via the ODE. Wang et al. (2022) specified the prior to measure how well $\hat{\mathbf{x}}'(t)$ fits the ODE system $\mathbf{f}$, along with a smoothing parameter λ to control the trade-off between the fit to (1.2) and the fidelity to (1.3):

$$\begin{aligned} \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \lambda) &\propto \exp \left\{ -\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\frac{d\hat{x}_i(t)}{dt} - f_i(\hat{\mathbf{x}}(t)|\boldsymbol{\theta}) \right]^2 dt \right\} \\ &= \exp \left\{ -\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\frac{d\Phi(t)^\top}{dt} \mathbf{c}_i - f_i(\mathbf{c}^\top \Phi(t)|\boldsymbol{\theta}) \right]^2 dt \right\}, \end{aligned} \quad (2.3)$$

where $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_I)$.

This procedure, however, involves the evaluation of the derivative $\hat{\mathbf{x}}'(t)$, which can be challenging and inefficient, as stated in Chen et al. (2017). When nonparametric smoothing is employed to estimate the derivative of an unknown function from noisy data, the convergence rate would be slower than that of estimating the function itself; see Stone (1985) and Theorem 3.6 of Fan and Gijbels (1995) for example. We take regression splines as an example to elaborate on this issue. B-spline basis functions are frequently used as the building blocks in regression splines. Without loss of generality, we assume that the unknown function is defined on $[0, 1]$, and that a large number of the standard B-spline basis functions with (almost) equally spaced knots are employed to approximate the function. Then such B-spline bases have locally compact support; i.e., these functions vanish in most subintervals defined by the knots. Given that they are bounded by 0 and 1, the basis functions increase from 0 to 1 rapidly. Consequently, their derivatives are bumpy, which explains why using regression splines to estimate the derivative of an unknown function from a relatively small number of noisy observations is challenging.

Thus, we consider a similar prior structure to (2.3) but without the derivative estimation. Rather than directly incorporating (1.3) into the prior, we integrate both sides of (1.3) with $\mathbf{x}(t)$ replaced by its basis approximation and measure the discrepancy between

$\hat{\mathbf{x}}(t)$ and the integral $\int \mathbf{f}(\hat{\mathbf{x}}(t), \boldsymbol{\theta}, t) dt$ (Dattner and Klaassen, 2015; Chen et al., 2017):

$$\begin{aligned}\tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \lambda) &\propto \exp \left\{ -\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\hat{x}_i(t) - \int_0^t f_i(\hat{\mathbf{x}}(s)|\boldsymbol{\theta}) ds - \hat{x}_i(0) \right]^2 dt \right\} \\ &= \exp \left\{ -\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\Phi(t)^\top \mathbf{c}_i - \int_0^t f_i(\mathbf{c}^\top \Phi(s)|\boldsymbol{\theta}) ds - \Phi(0)^\top \mathbf{c}_i \right]^2 dt \right\}.\end{aligned}\quad (2.4)$$

We shall treat λ as a tuning hyperparameter, as will be discussed in Section 2.2.3. To complete the posterior specification, it remains to choose prior distributions for $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$. Without specific prior information for the model parameters, we may simply assign the generic non-informative priors $\tilde{\pi}_0(\boldsymbol{\theta}) \propto 1$ and $\tilde{\pi}_0(\sigma_i) \propto 1/\sigma_i$ independently. Then the posterior distribution of $(\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma})$ for the inference is

$$\begin{aligned}\pi(\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma}|\mathbf{y}, \lambda) &\propto \pi(\mathbf{y}|\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma}, \lambda) \pi(\mathbf{c}|\boldsymbol{\theta}, \boldsymbol{\sigma}, \lambda) \tilde{\pi}_0(\boldsymbol{\theta}) \tilde{\pi}_0(\boldsymbol{\sigma}) \\ &= p(\mathbf{y}|\mathbf{c}, \boldsymbol{\sigma}) \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \lambda) \tilde{\pi}_0(\boldsymbol{\theta}) \tilde{\pi}_0(\boldsymbol{\sigma}),\end{aligned}\quad (2.5)$$

since in the factorization, $\mathbf{y}$ only depends on $\mathbf{c}$ and $\boldsymbol{\sigma}$ through (2.2), and $\mathbf{c}$ only depends on $\boldsymbol{\theta}$ and λ through (2.4).

Remark. If we assume an alternative distribution for measurement error ϵ_{ij} , the term $p(\mathbf{y}|\mathbf{c}, \boldsymbol{\sigma})$ in (2.5) should be replaced by the corresponding likelihood function.

2.2.2 Integral Estimation

The posterior distribution (2.5) involves two integrals that do not usually have closed-form expressions, so we employ numerical techniques to approximate them. For the outer integral, we apply the Gaussian quadrature rule (Stroud and Secrest, 1966) to the expansion,

$$\begin{aligned}\mathbf{R}_i &= \int_{t_1}^{t_J} \left[\Phi(t)^\top \mathbf{c}_i - \int_0^t f_i(\mathbf{c}^\top \Phi(s)|\boldsymbol{\theta}) ds - \Phi(0)^\top \mathbf{c}_i \right]^2 dt \\ &\approx \sum_{m=1}^{M_i} v_m \left[\Phi(\xi_m)^\top \mathbf{c}_i - \int_0^{\xi_m} f_i(\mathbf{c}^\top \Phi(s)|\boldsymbol{\theta}) ds - \Phi(0)^\top \mathbf{c}_i \right]^2,\end{aligned}\quad (2.6)$$

where M_i denotes the number of quadrature points used to evaluate the outer integral, $\{\xi_m\}_{m=1}^{M_i}$ are the quadrature points inside the interval $[t_1, t_J]$, and $\{v_m\}_{m=1}^{M_i}$ are the corresponding quadrature weights. For the inner integral, the Gaussian quadrature rule is also

applied to the B-spline expansion,

$$\mathbf{Q}_{m_i} = \int_0^{\xi_m} f_i(\mathbf{c}^\top \Phi(s) | \boldsymbol{\theta}) ds \approx \sum_{k=1}^{K_{m_i}} w_k f_i(\mathbf{c}^\top \Phi(s_k) | \boldsymbol{\theta}), \quad (2.7)$$

where K_{m_i} is the number of quadrature points used to evaluate the inner integral, $\{s_k\}_{k=1}^{K_{m_i}}$ are the quadrature points inside the interval $[0, \xi_m]$, and $\{w_k\}_{k=1}^{K_{m_i}}$ are the corresponding quadrature weights.

With K quadrature points, the Gaussian quadrature rule is exact for the integral of a function which can be well-approximated by a polynomial of degree $(2K - 1)$ or less (Golub and Welsch, 1969). For integrating products of B-splines with degree M between two adjacent knots, Gaussian quadrature needs $(M + 1)$ quadrature points to be exact (de Boor et al., 1976)³. Therefore, for the outer integral, the Gaussian quadrature is exact with $(M + 1)(L - 2)$ quadrature points, where $(L - 2)$ is the number of interior knots. In our simulation study (see Section 2.3), we use 81 interior knots, which require 324 quadrature points for the outer integral. However, when we use 200 quadrature points, the result is close to the exact integration. Using more quadrature points can improve the approximation accuracy, but also increases the computational cost. Thus as a general guideline in practice, we recommend using roughly $(M + 1)(L - 2)/2$ quadrature points for the outer integral and using the exact $(2K - 1)$ quadrature points for the inner integral.

2.2.3 The Choice of Smoothing Parameter

We now consider the choice of the smoothing parameter λ . Within a Bayesian framework, one approach is to assign a prior distribution to λ so that it is inferred along with the other components of the posterior distribution (Wang et al., 2022). A Gamma distribution was suggested as the prior for λ by adapting the work of Berry et al. (2002), but they did not provide a guideline on how to choose its hyperparameters. From the frequentist perspective, Ramsay et al. (2007) and Qi and Zhao (2010) recommended starting with a small λ and iteratively increasing its value until the parameter estimates become stable. Specifically, Ramsay et al. (2007) proposed to stop increasing λ once the norm of the difference between the ODE solution $\mathbf{x}(t | \boldsymbol{\theta}, \mathbf{x}_0)$ obtained from numerical solvers and the approximation $\hat{\mathbf{x}}(t)$ begins to increase after attaining a minimum; Qi and Zhao (2010) compared the

³If the function f is linear, its exact integral can be obtained easily. If f is nonlinear (e.g., polynomial), the number of quadrature points required for an exact integral can be known when using B-splines, but not when using a numerical solver (in terms of the adaptive step sizes and computational cost required).

ratio of overlaps of the confidence intervals of parameters for different λ . We synthesize these suggestions and propose Algorithm 1 for automatically choosing λ . Our algorithm begins with a small λ and increases it by a factor of 10 iteratively. The stopping rules of the algorithm consider the inference of parameter estimates, the discrepancy between the observations and the estimated ODE solution, and the discrepancy between the B-spline approximation and the estimated ODE solution. The final MCMC samples for inference of θ, c and σ are those associated with the output value $\hat{\lambda}$ chosen by Algorithm 1.

In practice, to assess stability in parameter estimation, we suggest choosing $\alpha = 0.1$ ⁴, and thus $1 - \alpha = 0.9$ as the threshold for the ratio of overlaps of the credible intervals of parameters for different λ . When the observation time points are relatively dense, Ramsay et al. (2007) suggested that a small λ may suffer distortion from using basis expansions to approximate the ODE solutions. The simulation results of Ramsay et al. (2007) and Qi and Zhao (2010) both indicate a relatively large λ can provide stable and accurate estimates for θ . Since we estimate the integral instead of the derivative in our framework, the case here is slightly different. However, the intuition is similar: when λ is small, the B-spline tends to fit the data better than the ODE model; then as λ is increased, the B-spline more closely approximates the underlying truth for the ODE model, until the stopping criterion is reached, i.e., a stable inference result for θ or a minimal discrepancy in (2.8) is attained. Based on their intuitions, we recommend starting at $\lambda^{(0)} = 10^2$ with $\lambda^* = 10^{35}$. To achieve a better trade-off between fidelity to data and fitting the ODE system for relatively sparse observations, we recommend $\lambda^{(0)} = 1$ and $\lambda^* = 10^2$.

Remark. When λ satisfies $\lambda/J^2 \rightarrow \infty$ as J diverges and some other regularity conditions are met, Qi and Zhao (2010) established large sample properties for the parameter estimator and the estimated solution under the generalized profiling framework proposed by Ramsay et al. (2007). Thus we borrow their idea of leveraging asymptotic confidence intervals to select λ . However, as pointed out by one referee, an approximation error is introduced when using integration to replace the differentiation as it cannot honestly represent the constraint from the original ODEs. Therefore, we take three factors into consideration when selecting λ : inference for the parameters, fit to (1.2) and approximation error induced by the spline representation. Our numerical studies demonstrate selection consistency of λ of our rule and the asymptotic behavior of the estimated parameters and ODE solutions; see Section B.1.4 in the appendix for more details.

⁴ α is the level for the overlap ratio of the credible interval in Algorithm 1.

⁵ λ^* denotes the threshold after which the stopping criterion starts to be checked in Algorithm 1.

Algorithm 1 Smoothing parameter algorithm

- 1: **Inputs:** a small initial $\lambda^{(0)}$; a moderately large threshold λ^* after which the stopping criterion starts to be checked; the level α for the overlap ratio of the credible interval; initial estimates $\hat{\boldsymbol{\theta}}(\lambda^{(0)})$, $\hat{\mathbf{c}}(\lambda^{(0)})$ and $\hat{\boldsymbol{\sigma}}(\lambda^{(0)})$ corresponding to $\lambda^{(0)}$
- 2: **Output:** the final choice $\hat{\lambda}$
- 3: Set $p = 1$, $\lambda^{(1)} = \lambda^{(0)} \times 10$
- 4: **while** $\lambda^{(p)} \leq 10^6$ **do**
- 5: Sample $\boldsymbol{\theta}$, $\mathbf{c}$ and $\boldsymbol{\sigma}$ from the posterior distribution $\pi(\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma} | \mathbf{y}, \lambda^{(p)})$ given by (2.5), using the values $\hat{\boldsymbol{\theta}}(\lambda^{(p-1)})$, $\hat{\boldsymbol{\sigma}}(\lambda^{(p-1)})$ and $\hat{\mathbf{c}}(\lambda^{(0)})$ to initialize the MCMC sampler
- 6: Using the MCMC samples, compute the posterior means $\hat{\boldsymbol{\theta}}(\lambda^{(p)})$, $\hat{\mathbf{c}}(\lambda^{(p)})$, $\hat{\boldsymbol{\sigma}}(\lambda^{(p)})$ for $\boldsymbol{\theta}$, $\mathbf{c}$ and $\boldsymbol{\sigma}$, and the central 95% credible interval for each parameter in $\boldsymbol{\theta}$
- 7: Calculate

$$\begin{aligned}
Err(\lambda^{(p)}) = & \sum_{i=1}^I \left\{ \sum_{j=1}^J \left[y_{ij} - \int_0^{t_j} f_i(\hat{\mathbf{c}}(\lambda^{(p)})^\top \Phi(s) | \hat{\boldsymbol{\theta}}(\lambda^{(p)})) ds - \Phi(0)^\top \hat{\mathbf{c}}_i(\lambda^{(p)}) \right]^2 \right. \\
& \left. + \left[\Phi(t_J)^\top \hat{\mathbf{c}}_i(\lambda^{(p)}) - \int_0^{t_J} f_i(\hat{\mathbf{c}}(\lambda^{(p)})^\top \Phi(s) | \hat{\boldsymbol{\theta}}(\lambda^{(p)})) ds - \Phi(0)^\top \hat{\mathbf{c}}_i(\lambda^{(p)}) \right]^2 \right\}, \quad (2.8)
\end{aligned}$$

which measures the discrepancies between the observations and the estimated ODE solution, and between the B-spline approximation and the estimated ODE solution. Here, $\int_0^t f_i(\hat{\mathbf{c}}(\lambda^{(p)})^\top \Phi(s) | \hat{\boldsymbol{\theta}}(\lambda^{(p)})) ds$ is approximated with the Gaussian quadrature as discussed in Section 2.2.2.

- 8: **if** $\lambda^{(p)} \geq \lambda^*$ **then**
 - 9: **if** $Err(\lambda^{(p)}) \leq Err(\lambda^{(p-1)})$ **then**
 - 10: For each parameter in $\boldsymbol{\theta}$, calculate the overlap of its credible intervals corresponding to $\lambda^{(p)}$ and $\lambda^{(p-1)}$
 - 11: **if** the ratio of the overlap to the current credible interval for each parameter in $\boldsymbol{\theta}$ is larger than $1 - \alpha$ **then**
 - 12: **return** $\hat{\lambda} = \lambda^{(p)}$
 - 13: **end if**
 - 14: **else if** $Err(\lambda^{(p)}) > Err(\lambda^{(p-1)})$ **then**
 - 15: **return** $\hat{\lambda} = \lambda^{(p-1)}$
 - 16: **else if** $\lambda^{(p)} = 10^6$ **then**
 - 17: **return** $\hat{\lambda} = \lambda^{(p)}$
 - 18: **end if**
 - 19: **end if**
 - 20: $\lambda^{(p+1)} = \lambda^{(p)} \times 10$
 - 21: $p \leftarrow p + 1$
 - 22: **end while**
-

2.2.4 Practical Implementation

We implement the posterior distribution (2.5) in C++ and pass it to *tmbstan* (Monnahan and Kristensen, 2018). NUTS (Homan and Gelman, 2014) is used as the sampling algorithm, which enables a more efficient exploration of the posterior distribution compared to traditional MCMC samplers. One chain of 400⁶ iterations, including a 200-iteration warmup, is run for each $\lambda^{(p)}$ in our algorithm. More implementation details can be found in Sections A.1 and B.2 of the appendix.

Regarding the initial estimates, we first use the R package *CollocInfer* (Hooker et al., 2016) to smooth the data using the roughness penalty 0.1 and extract the coefficients as $\hat{\mathbf{c}}(\lambda^{(0)})$ ⁷. With other parameters fixed, we set $\hat{\sigma}_i(\lambda^{(0)}) = 0.1$ and then plug $\hat{\mathbf{c}}(\lambda^{(0)})$ into (2.4). Using the draws from one 400-iteration chain, including a 200-iteration warmup, the posterior mean of $\boldsymbol{\theta}$ is treated as the initial estimate $\hat{\boldsymbol{\theta}}(\lambda^{(0)})$. The posterior means of the sample are used as the estimates for the parameters.

2.3 Simulation Study

2.3.1 Benchmark System and Setup

In this simulation study, we evaluate our method using the FitzHugh–Nagumo (FN) equations as a benchmark system (FitzHugh, 1961). The FN system is a set of two coupled non-linear differential equations describing the behaviour of spike potentials, which take the form

$$\mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} c \left(V - \frac{V^3}{3} + R \right) \\ -\frac{1}{c}(V - a + bR) \end{bmatrix}, \quad (2.9)$$

where $\mathbf{x}(t) = (V(t), R(t))^\top$, V denotes the voltage of the neuron membrane potential, R is the recovery variable from neuron currents, and the parameter vector is $\boldsymbol{\theta} = (a, b, c)$. As in Yang et al. (2021), we set the true values of the parameters as $\boldsymbol{\theta} = (0.2, 0.2, 3)$ and the initial conditions as $V(0) = -1$ and $R(0) = 1$. For both V and R , noisy observations at different time points are generated with noise level $\boldsymbol{\sigma} = (\sigma_1, \sigma_2) = (0.2, 0.2)$. To demonstrate

⁶The small number of iterations is due to computational challenges, and a more efficient MCMC implementation is left for future work.

⁷*CollocInfer* is used only for the initialization of smoothing spline estimation for $x_i(t)$ based on observations; it is not applied to ODE parameter estimation.

inference results without specific prior knowledge of the parameters, we set the priors for a, b, c to be uniform over positive real numbers.

To investigate the effect of different sampling frequencies, i.e., the number or denseness of observation time points, we first generate 41 and 21 equally-spaced noisy observations of V and R on the time interval $[0, 20]$, which follow the same settings in [Yang et al. \(2021\)](#). For these two cases, we place 81 equally-spaced knots with 83 basis functions within the time interval. When implementing the smoothing parameter selection algorithm, we start from $\lambda^{(0)} = 10^2$ with $\lambda^* = 10^3$. To further explore the performance of our method with extremely sparse observations, we also consider the case with only 11 observations. In this case, we consider 23 basis functions with 21 equally spaced knots, and choose $\lambda^{(0)} = 1$ and $\lambda^* = 10^2$. As described in Section 2.2.2, we use Gaussian quadrature to estimate the integrals in (2.6) and (2.7). Following the suggested guidelines, for the outer integral in (2.6), we choose $M_i = 200$ quadrature points for 41 and 21 observations and $M_i = 50$ for 11 observations within $[0, 20]$. Since the FN system contains a polynomial of degree 3, $K_{m_i} = 5$ quadrature points are chosen for each interval $[0, \xi_m]$ for the inner integral in (2.7).

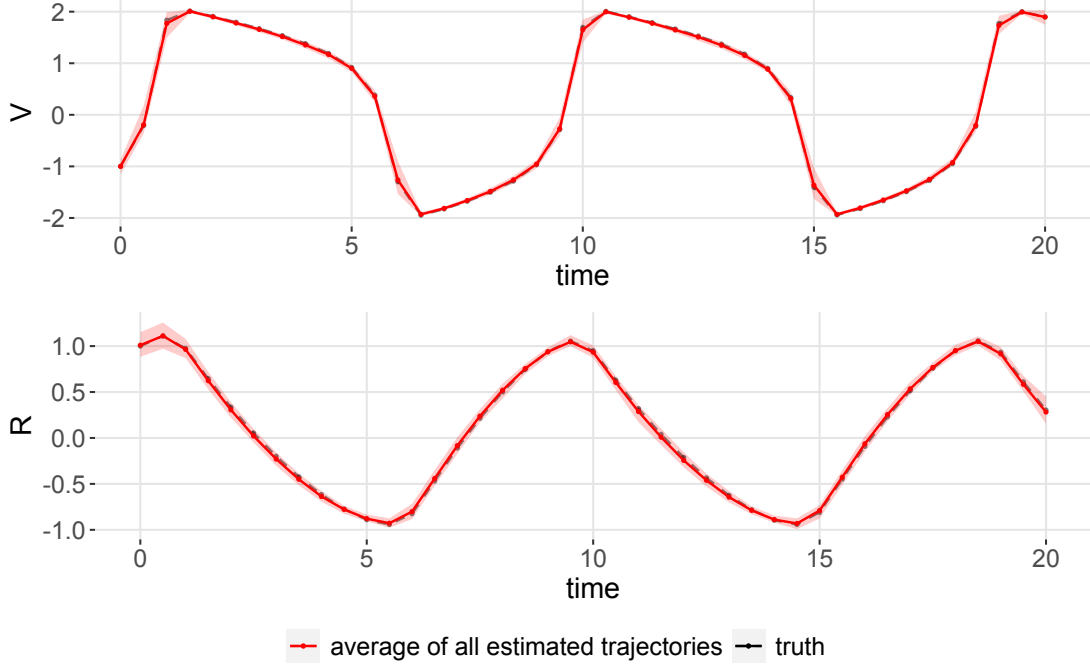


Figure 2.1: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 41 observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

To assess the performance of our method, we compare it with three other Bayesian methods⁸: manifold-constrained Gaussian process inference (MAGI, [Yang et al., 2021](#)), fast Gaussian process-based gradient matching (FGPGM, [Wenk et al., 2019](#)), and a Bayesian collocation method that replaces prior (2.4) containing only integrals with prior (2.3) containing derivatives. For the two GP-based methods, we strictly follow the recommendation of the authors. Particularly, to implement MAGI, we employ their R package *magi*: 161 and 321 discretizations are used to approximate the random variable that quantifies the difference between GP derivatives and ODE systems for 41 and 21 observations, respectively. In the case of 11 observations, the number of discretizations is set to 161. Additionally, since the default estimation for hyperparameter ϕ may be unreliable in this case, any value less than 0.5 in ϕ is overridden by 0.5. 20,000 Hamiltonian Monte Carlo (HMC, [Neal, 2011](#))

⁸The main focus of this paper is the comparison of our method with other Bayesian methods, hence methods such as *CollocInfer* were not included when comparing estimation performance.

iterations are run, including a 10,000-iteration warmup. The number of leapfrog steps per HMC iteration is set to 100. For FGPGM, we run their software: the standard deviation γ that adjusts the potential model mismatch between the ODE output and GP derivatives is set to 3×10^{-4} , along with a Matérn52 kernel. 300,000 MCMC iterations are run, including the first half of the iterations as a warmup. For the Bayesian collocation method with derivative estimation, we consider exactly the same setting as our method described in Section 2.2.4. All these comparison methods take the posterior means of the sample as the parameter estimates. To evaluate the variability of the parameter estimates, 100 independent simulation trials are run. In all simulation studies, the R package *deSolve* (Soetaert et al., 2010) is used for numerically solving the differential equations.

2.3.2 Performance Assessment

We consider two metrics to assess the performance of the parameter estimates and their associated trajectories. For the parameter estimates, we calculate the root mean square error (RMSE) to the true parameter value. The mean of each parameter estimate is also reported for comparing the bias. Regarding the system recovery, Yang et al. (2021) proposed a trajectory RMSE metric to assess the performance in recovering systems. To fairly compare simulation results for the same ODE system under different sampling frequencies, we adapt their metric by defining

$$\text{trajectory RMSE} = \sqrt{\int_{t_1}^{t_J} [\tilde{x}(t) - x(t)]^2 dt},$$

where $\tilde{x}(t)$ and $x(t)$ denote the reconstructed and the true trajectories, respectively. We approximate the integral inside via the Riemann sum with a dense set of grid points. In summary, the trajectory RMSEs are calculated as follows: first, we apply a numerical solver to the ODE model (1.3), with f and θ given by (2.9) and the estimated parameters respectively, to reconstruct the trajectory implied by those estimates; second, we likewise numerically solve the true trajectories $\mathbf{x}(t)$ based on the true parameters and initial conditions; then, we choose a sufficiently large number of time points in the observation time interval $[t_1, t_J]$; lastly, we calculate the RMSE of the reconstructed trajectories $\tilde{\mathbf{x}}(t)$ to the true trajectories $\mathbf{x}(t)$ using these time points. It should be noted that a numerical solver is employed to calculate the trajectory RMSE only for performance assessment, while it is not required for implementing our method. To better understand the magnitude of the error in recovering the trajectories, we also report the average norm of each estimated component under each simulation setting in Section B.1 of the appendix.

2.3.3 Results

Observations	Method	a	b	c
41	Integral	0.20(0.02)	0.26(0.10)	2.95(0.07)
	Derivative	0.20(0.03)	0.28(0.12)	2.75(0.29)
	MAGI	0.20(0.02)	0.33(0.16)	2.89(0.12)
	FGPGM	0.22(0.05)	0.32(0.19)	2.88(0.20)
21	Integral	0.20(0.03)	0.37(0.21)	2.86(0.16)
	Derivative	0.20(0.03)	0.31(0.17)	2.52(0.55)
	MAGI	0.19(0.03)	0.44(0.28)	2.79(0.25)
	FGPGM	0.25(0.10)	0.19(0.16)	2.69(0.47)
11	Integral	0.14(0.09)	0.69(0.49)	1.35(1.69)
	Derivative	0.07(0.13)	0.64(0.44)	0.95(2.05)
	MAGI	0.11(0.11)	0.66(0.46)	1.03(1.97)
	FGPGM	0.30(0.23)	0.30(1.10)	0.55(2.76)

Table 2.1: Mean of the parameter estimates with RMSE in parentheses for the FN system across 100 simulated runs.⁹

Table 2.1 summarizes the results of parameter estimation for the four methods under different sampling frequencies, while the median and the interquartile range (IQR) of trajectory RMSEs are presented in Table 2.2. Furthermore, Section B.1.1 of the appendix provides the boxplots of parameter and trajectory RMSEs for these methods. Under the setting of 41 noisy observations, our proposed method has the lowest RMSEs and biases among all the methods when estimating θ . Despite a few outliers with higher trajectory RMSEs than MAGI of component V as shown in Figure B.4, our method has the lowest trajectory RMSEs when reconstructing the ODE solution. The derivative method generally has higher parameter and trajectory RMSEs than MAGI and FGPGM. Moreover, the system component V has a higher trajectory RMSE than the component R for each method. The differences in trajectory RMSEs between our method and the derivative method seem to be smaller for R than those for V , indicating that evaluating the derivative can be more challenging and inefficient for nonlinear functions.

⁹NOTE: Bold values indicate the lowest RMSE in each setting.

Observations	Method	V	R	Total
41	Integral	0.086(0.055)	0.045(0.040)	0.104(0.057)
	Derivative	0.235(0.170)	0.115(0.084)	0.268(0.177)
	MAGI	0.120(0.057)	0.083(0.043)	0.148(0.061)
	FGPGM	0.224(0.184)	0.072(0.072)	0.251(0.196)
21	Integral	0.165(0.085)	0.098(0.073)	0.196(0.105)
	Derivative	0.822(0.614)	0.402(0.306)	0.915(0.645)
	MAGI	0.173(0.071)	0.136(0.085)	0.219(0.093)
	FGPGM	0.596(0.408)	0.202(0.217)	0.630(0.454)
11	Integral	0.961(0.240)	0.255(0.280)	0.996(0.244)
	Derivative	1.051(0.192)	0.197(0.110)	1.077(0.166)
	MAGI	1.045(0.214)	0.200(0.092)	1.060(0.190)
	FGPGM	1.461(0.002)	0.828(0.089)	1.686(0.033)

Table 2.2: Median of trajectory RMSEs (with IQR in parentheses) for each component of the FN system across the 100 simulated runs.¹⁰

To showcase the performance of our method in estimating the ODE solution, we consider the estimated $x_i(t)$ based on the spline approximation (2.1), i.e., $\Phi(t)^\top \hat{\mathbf{c}}_i$, rather than a numerical solver. Figure 2.1 displays the average of these estimates across 100 simulation runs as well as pointwise central 95% intervals under the setting of 41 observation time points. This figure indicates that our method not only recovers the system reasonably well, but also provides reliable inference for the ODE solution. Furthermore, we examine the estimated $x_i(t)$ with 95% credible intervals from a randomly selected simulation run, and the plots are presented in Section B.1.1 of the appendix.

¹⁰NOTE: Bold values indicate the lowest RMSE in each setting.

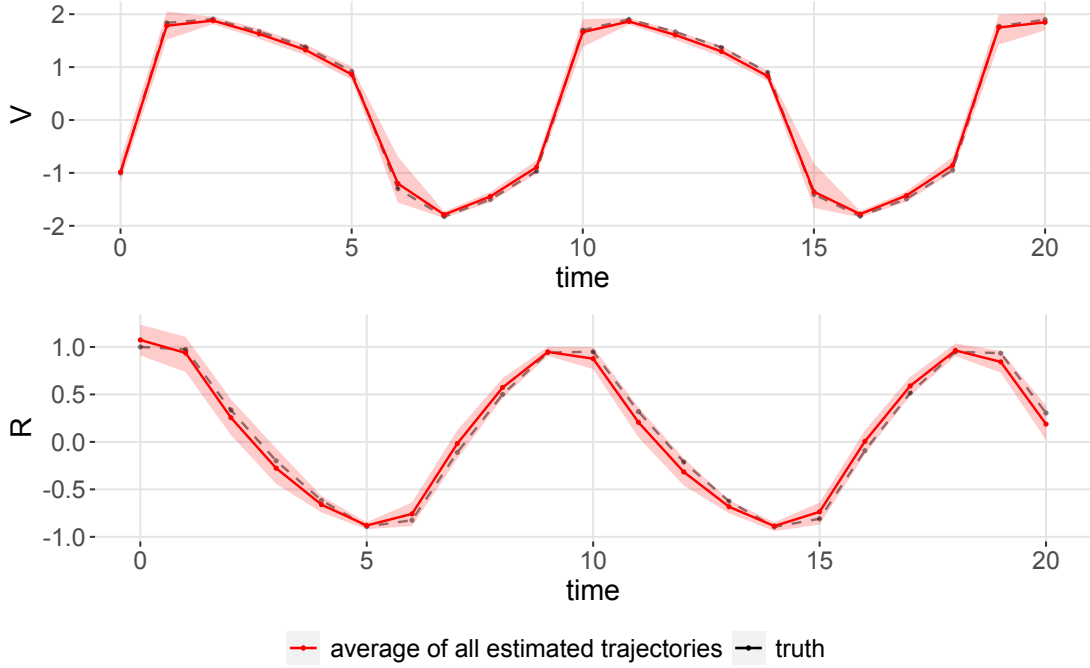


Figure 2.2: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 21 observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

Under the setting of 21 observations, our method still achieves the lowest parameter RMSEs and biases among all methods, except that the parameter b has a higher RMSE than the derivative method and FGPGM, as displayed in Table 2.1 and Figure B.2 in the appendix. From Figure B.5, we notice that the boxplot of our method appears to have outliers with significantly higher trajectory RMSEs than MAGI for V and R . Despite this, our method still outperforms the competitors in terms of the trajectory RMSEs, while the derivative method is the worst, as shown in Table 2.2. Similar to the case of 41 observations, the trajectory RMSE of V is higher than that of R for all methods. When comparing our method with the derivative method, the discrepancy between trajectory RMSEs of V is considerably larger than that of R . Figure 2.2 presents the average of the estimated trajectories of our method across 100 simulation runs as well as the 95% pointwise intervals. Though the 95% pointwise intervals are slightly wider than those under the setting of 41 observations, our method can still accurately estimate the ODE solution.

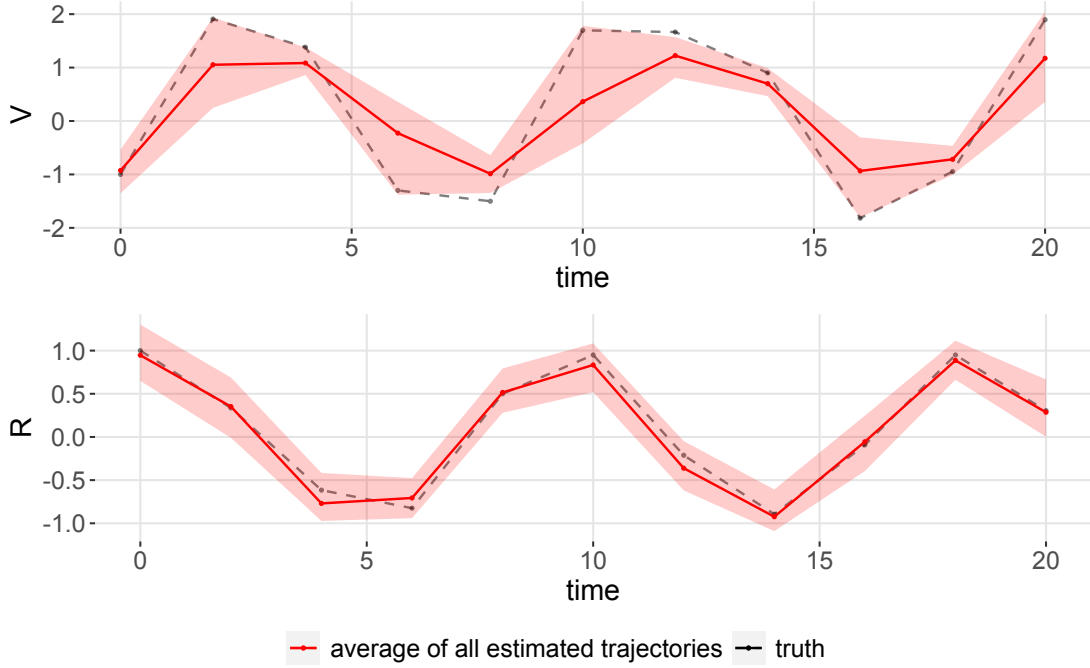


Figure 2.3: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 11 observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

In the case of 11 observations, all methods suffer from poor parameter estimation as expected, and the parameter RMSEs are significantly higher than with 41 or 21 observations, as reported in Table 2.1 and Figure B.3 in the appendix. However, our method is still superior to the other three competitors. Although the derivative method has the lowest RMSE for parameter b , it has higher RMSEs and biases than our method for a and c . Table 2.2 and Figure B.6 show that all methods fail to recover the ODE solution accurately. Compared with other methods, our proposed method has a marginally lower total trajectory RMSE. Furthermore, Table 2.2 indicates that estimating V is much more challenging than estimating R for all methods. In particular, when comparing our method with the derivative method, the trajectory RMSEs of component R are similar, whereas there exists a remarkable difference in those of component V . Figure 2.3 displays the average of the estimated ODE solution across 100 simulation runs as well as the 95% pointwise intervals. In contrast to R , the true trajectory V is not fully covered by the 95% pointwise intervals. Moreover, Figure 2.3 shows that the discrepancy between $\hat{V}$ and V is considerably larger

than that between $\hat{R}$ and R . This finding further justifies the difficulty of estimating V due to the nonlinear structure in (2.9), especially under the setting of sparse observations.

We also compare computational time of each method when fitting the FN system. Our method is considerably more efficient than MAGI and FGPGM in computations. More details about this comparison can be found in Section B.1.1 of the appendix.

Moreover, we investigate the applicability of our method to irregularly spaced observations via two additional simulation studies: one is the FN system with 11 unequally-spaced noisy observations at $t = \{0, 1, 2, 3, 6, 9, 12, 15, 18, 19, 20\}$ (the rest of the setup is identical to the setup of Section 2.3.1); the other is the protein transduction system with unidentifiable parameters studied by Yang et al. (2021). The overall conclusions of these two simulation studies are similar to our findings for the FN system with equally-spaced observations, and more details are provided in Sections B.1.2 and B.1.5 of the appendix.

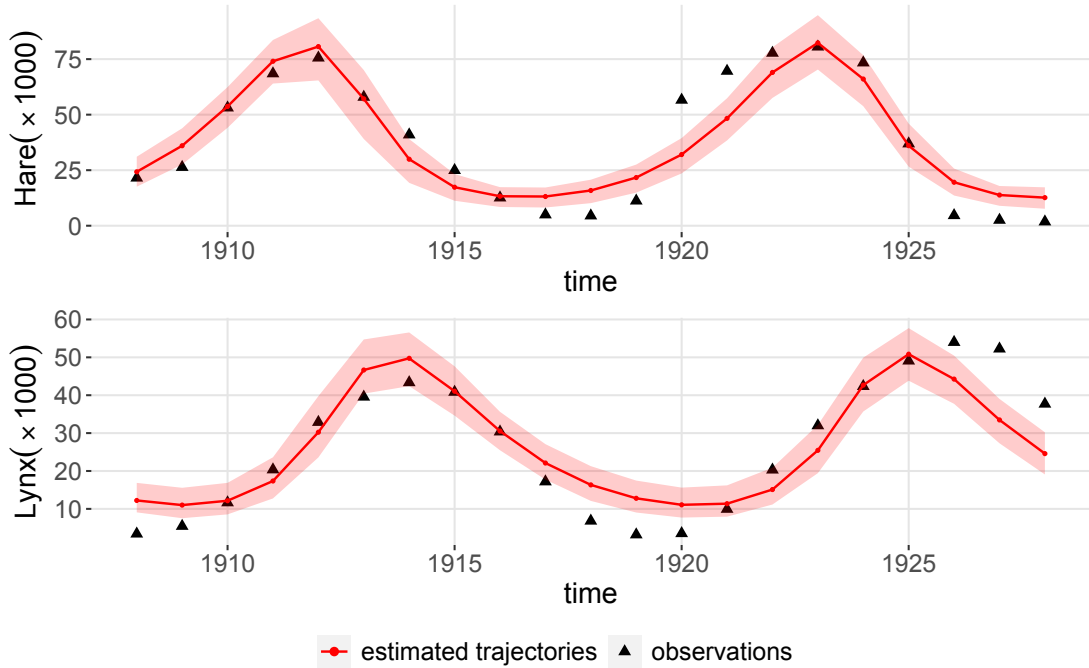


Figure 2.4: Estimated trajectories for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

2.4 Real Data Analysis

Snowshoe hares and Canadian lynxes are generally believed to be related, since lynxes are specialist predators of hares. During the 1800s and 1900s, the Hudson’s Bay company, which was the largest fur trapper in Canada, kept careful records of the numbers of these two animals that had been traded. [Odum \(1953\)](#) provided the data for 1845-1935. Figure [2.4](#) (black triangles) displays the distinct oscillations of these two populations for 1908-1928. Specifically, when the lynx population is sufficiently low, the hare population grows, which allows the lynx population to grow and become large enough to cut down on the hare population. A decline in the hare population would lead to a shrinkage of the lynx population and end one cycle of this interaction. To explain this population fluctuation, [May \(1973\)](#) suggested using the Lotka-Volterra (LV) model ([Lotka, 1932](#)) to fit the data.

The LV model describes the interaction between the population of the prey (x_1) and that of the predator (x_2) over time. The system consists of two equations:

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{pmatrix} \theta_1 x_1 - \theta_2 x_1 x_2 \\ -\theta_3 x_2 + \theta_4 x_1 x_2 \end{pmatrix}, \quad (2.10)$$

where the parameter vector is $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)$. Here x_1 and x_2 in [\(2.10\)](#) represent the numbers of snowshoe hares and Canadian lynxes, respectively.

Without specific prior information, the prior distributions for each parameter in $\boldsymbol{\theta}$ are set to be uniform on $(0, \infty)$. We take 43 cubic B-spline basis functions with 41 knots equally spaced over $[1908, 1928]$. To select the smoothing parameter λ , we let Algorithm [1](#) start at $\lambda^{(0)} = 1$ with $\lambda^* = 10^2$. In accordance with the guideline described in Section [2.2.2](#), we take $M_i = 100$ and $K_{m_i} = 4$ quadrature points to approximate the outer integral [\(2.6\)](#) and inner integral [\(2.7\)](#), respectively. The other settings for implementing our method are taken to be the same as in the simulation studies. Furthermore, we report the implementation details and results of MAGI, FGPGM, and the Bayesian collocation method with derivative estimation on this dataset in Section [B.2](#) of the appendix.

Using Algorithm [1](#) introduced in Section [2.2.3](#), $\hat{\lambda} = 10$ is selected. Table [2.3](#) summarizes the parameter estimation results for model [\(2.10\)](#), and Figure [2.4](#) depicts the estimated trajectories based on our method. As real time-course observations are inherently noisy, the 95% pointwise credible intervals from our fitted LV model do not completely cover all the points. Nonetheless, the estimated trajectories and 95% pointwise credible intervals provide reasonable fits for the time-varying numbers of lynxes and hares.

	Estimate
θ_1	0.720 (0.552, 0.971)
θ_2	0.028 (0.022, 0.036)
θ_3	0.496 (0.360, 0.659)
θ_4	0.013 (0.010, 0.017)
x_{01}	24.270 (17.559, 31.060)
x_{02}	12.220 (9.076, 16.865)

Table 2.3: Parameter estimates with 95% credible interval in parentheses, when using the LV model (2.10) to fit the hare and lynx data for 1908-1928. Here x_{01} and x_{02} denote the estimated initial conditions for x_1 and x_2 , respectively.

2.5 Discussion

In this paper, we develop a Bayesian collocation framework for parameter inference in ODE systems. In contrast to previous approaches, our work involves estimating an integral rather than a derivative. Moreover, our proposed method can be applied to general nonlinear ODEs and can select the smoothing parameter automatically. Simulation studies demonstrate that the proposed integral-based method outperforms derivative-based methods, including the existing GP-based approaches, in terms of inference accuracy. Furthermore, our method is considerably more efficient than GP-based methods.¹¹

We examine the stability of our method under different sampling frequencies in simulation studies (Section 2.3). As the number of observations decreases, our method yields wider pointwise intervals for estimated trajectories, and the performance of parameter estimates and recovered trajectories in the ODE system becomes worse for all methods. Compared with the Bayesian collocation derivative-based method, our method tends to recover nonlinear systems more effectively regardless of the sampling frequency. Furthermore, the difference in the performance of estimating parameters and recovering trajectories between these two methods also depends on the sampling frequency. We find a phase transition when comparing them under different sampling frequencies. In particular, when observations are dense enough to generate a reliable estimate for the derivatives, there exist no distinct differences between them, but the integral-based method is still superior. As the observations become sparser, the derivatives cannot be estimated that accurately. Consequently, the advantage of the integral-based method becomes particularly evident, especially in recovering the trajectories. With extremely sparse observations, only a minor

¹¹This manuscript was submitted on April 4th, 2023; the method comparisons may now be outdated.

difference is displayed between these two methods. A plausible reason is that recovering the system accurately is challenging for any method in this scenario.

We have also confirmed the inefficiency of methods that require numerical integration of the ODEs. We implemented a Bayesian method that is similar to NLS in *Stan* (Stan Development Team, 2022), which has the same priors for σ and θ as our method. Specifically, at each iteration, the method first proposes a new state for $\hat{\theta}$, $\hat{\mathbf{x}}_0$ and $\hat{\sigma}$, and then numerically integrates the ODE system using the Dormand-Prince algorithm, i.e., a fourth/fifth order Runge-Kutta method, to obtain the ODE solution. Under the same setup as in the simulation study, NUTS is used to sample four parallel chains of 1,000 iterations, using the first half of the iterations as a warmup. It turns out that for some simulated data sets, one or two chains fail to converge within the maximum allotted iterations. Even if we run more iterations, the parameter estimates still cannot converge toward their true values. As stated in Campbell (2007), the initial condition $\mathbf{x}_0$ determines the phase of oscillations of the trajectories in the FN system. If the estimated $\mathbf{x}_0$ is far from its true value, then trajectories would not even oscillate within the observed time interval. Consequently, the estimated θ could deviate significantly from its true value. Moreover, we find that due to the use of numerical solvers, this method could be extremely slow (i.e., on the order of days to complete 1,000 iterations) if the ODE system is stiff at the value of these parameters being sampled. In contrast, our method integrates the B-spline approximation using the Gaussian quadrature and does not suffer from these numerical solver issues; each example in the paper completes in the order of minutes.

Finally, our method might be limited to systems without completely unobserved variables. We leave an extension of our work to systems with entirely unobserved variables to further research.

Chapter 3

A Bayesian Collocation Integral Method for System Identification of Ordinary Differential Equations

3.1 Introduction

For high-dimensional time-course data analysis, many classes of models have been developed to characterize the dynamics of complex systems in various scientific areas, including gene regulatory networks, infectious diseases and neuroscience. These include state-space models (Rangel et al., 2004; Beal et al., 2004), regression models (Gupta et al., 2007), differential equation models (Lu et al., 2011; Wu et al., 2014; Chen et al., 2017; Dai and Li, 2022; Zhang et al., 2022) and graphical models (Abegaz and Wit, 2013; Liang et al., 2015). For high-dimensional complex systems, we might be uncertain about the functional relationships among variables; therefore, we assume $\mathbf{f}$ in (1.1) is unknown to provide greater flexibility. ODE models are commonly used to systematically describe both positive and negative feedback between variables (Wu et al., 2014). Moreover, they can naturally generate predictions for the variables at future times.

Then, the problem of interest can be to estimate $\mathbf{f}$, i.e., identifying the ODE system structure, from the observed data $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_J)$ with measurement model (1.2). As a result, scientists can analyze practical issues based on the reconstructed systems. For example, understanding the structure of gene regulatory networks can shed light on diseases caused by dysregulated cellular processes (Karlebach and Shamir, 2008). In particular, Rangel et al. (2004) demonstrated that their reconstructed network reflects many of the

dynamics associated with an activated T-cell, including the integrated activation of cytokines, proliferation, and adhesion following activation. For social science studies, [Lam et al. \(2023\)](#) identified resource coordination and information exchange by stakeholders in the inter-organizational network through social network analysis. In the high-dimensional setting for (1.1), only a small number of variables x_p , $p = 1, \dots, I$ are expected to control the change of x_i through its derivative x'_i for $i = 1, \dots, p$. Therefore, sparsity-inducing penalties are widely adopted to estimate $\mathbf{f}$ for such sparse ODE structures. Numerous efforts have been devoted to modelling high-dimensional sparse ODEs under various assumptions. One popular category is the two-step collocation method: first, spline estimates are fitted to the observations using smoothing methods; then, the unknown $\mathbf{f}$ is estimated given these estimates, along with a sparsity-inducing penalty. For example, assuming that each f_i in model (1.1) can be represented as a linear combination of x_p 's, [Lu et al. \(2011\)](#) proposed the smoothly clipped absolute deviation approach for variable selection. [Wu et al. \(2014\)](#) argued that the linear assumption can be restrictive for practical applications. Hence, they relaxed the assumption to be a more general additive structure and proposed an estimation procedure coupled with the adaptive group lasso technique ([Wang and Leng, 2008](#)). [Chen et al. \(2017\)](#) noted that the aforementioned methods entail estimation of the derivatives of ODE solutions, which is challenging as the performance can be highly sensitive to the accuracy of the estimated derivatives. To address this issue, they proposed a method of graph reconstruction via additive differential equations (GRADE), which integrates both sides of (1.1) under the same additive model assumption and utilizes the adaptive group Lasso penalty to select non-vanishing components in each f_i . Following this idea, [Dai and Li \(2022\)](#) proposed a class of kernel ODE models, which further relaxes the additive assumption to allow pairwise interactions in each f_i . Both [Chen et al. \(2017\)](#) and [Dai and Li \(2022\)](#) demonstrate that this integral-based technique leads to more robust estimates than derivative-based methods.

As an alternative to these two-step collocation methods, [Zhang et al. \(2022\)](#) proposed a joint estimation approach for generalized sparse additive ODEs (JADE), which accounts for the likelihood function of data, fidelity to the ODE structure, and sparse regularization simultaneously. They carried out extensive numerical studies to demonstrate that JADE can have better performance than two-step collocation methods in the estimation of trajectories and identification of sparse system structures. Nonetheless, all these aforementioned methods address the problem of interest from a frequentist perspective. In contrast, Bayesian approaches, which may better quantify the uncertainty in parameter estimation in such models ([Wang et al., 2022](#)), have yet to be well developed.

In this paper, we propose to improve on previous approaches by incorporating integral estimation into a Bayesian hierarchical collocation method. Under the additive assumption,

we consider using the expansion of basis functions such as cubic B-splines to approximate both $\mathbf{x}(t)$ and $\mathbf{f}(\mathbf{x}(t), t)$, which allows for these B-spline approximations to be integrated. Inspired by [Zhang et al. \(2022\)](#), for trajectory estimation, we combine the integrated ODE constraints and the likelihood of the observed data via a smoothing parameter to control the trade-off between the fit to the data and the fidelity to the ODEs. To identify the sparse additive ODE structure, we utilize a self-adaptive group-wise sparse penalty ([Bai et al., 2022](#)) to select the non-vanishing functional components in each f_i automatically. Proper priors are chosen for each relevant parameter to formulate the joint likelihood, and then a Markov chain Monte Carlo (MCMC) sampler is used to draw samples from the posterior distribution. Compared with the derivative-based methods, our integral-based method can improve the performance in system reconstruction and trajectory estimation, especially when data are not densely observed. Regarding the statistical inference of the estimand, the uncertainty of estimates can be easily quantified under our Bayesian framework. In particular, the proposed method allows us to compute the pointwise coverage probabilities for the ODE trajectories, whereas the existing frequentist methods are challenging to implement for this purpose. Such ODE uncertainty quantification can be crucial for scientists to ensure the reliability of their conclusions. In RNA-seq experiments, quantifying gene expression levels in individual samples or identifying differentially expressed genes between samples can involve inherent uncertainty arising from library preparation and ambiguities in read mapping ([Mandric et al., 2017](#)). In this context, [Hsu et al. \(2006\)](#) noted that constructing confidence intervals is key to both confirming the existence of differential expressions and bounding the magnitude of such differences. Moreover, confidence or credible bands on the estimated trajectories of expression levels over time can help validate whether dynamic behaviors such as oscillations or steady states reflect meaningful trends rather than random noise ([St John and Doyle III, 2013](#)). We demonstrate the favorable performance of our method for identifying and recovering the sparse ODE structure as well as its capability of quantifying uncertainty in trajectory estimation through simulation studies.

The remainder of the paper is organized as follows. Section [3.2](#) provides the formulation of our Bayesian method, including a detailed description of the algorithm and computational techniques. The performance of the proposed method is investigated in Section [3.3](#) and Section [3.4](#) using simulated and real data, respectively. Finally, we conclude the paper with some discussions in Section [3.5](#). Some implementation details and additional numerical results are relegated to the Section [C](#) in the appendix.

3.2 Methodology

3.2.1 Bayesian Structure

Under the additive assumption, the ODE system (1.1) takes the form (Wu et al., 2014)

$$x'_i(t) = a_{i0} + \sum_{p=1}^I f_{ip}\{x_p(t)\}, \quad i = 1, 2, \dots, I, \quad (3.1)$$

where a_{i0} is the intercept term and $f_{ip}\{x_p(t)\}$ is the unknown additive component, which characterizes the effect of x_p on the derivative of x_i . Due to the sparseness principle of gene regulatory networks and other systems, in practice, the number of non-zero f_{ip} 's is usually assumed to be small (Wu et al., 2014; Chen et al., 2017). Additionally, the number of variables I generally exceeds the number of observation times J in time-course observations. In this paper, we refer to (3.1) as the sparse additive ODE model. Note that the additive components are not fully identifiable without extra assumptions. For example, if we add a constant $r \in \mathbb{R}$ to one f_{ip} and subtract r from another, the ODE system will remain the same. This issue can be addressed by imposing the following constraint on each f_{ip} (Chen et al., 2017; Dai and Li, 2022):

$$\int_{t_1}^{t_J} f_{ip}\{x_p(t)\} dt = 0, \quad i, p = 1, \dots, I. \quad (3.2)$$

Following Zhang et al. (2022), we approximate $f_{ip}\{x_p(t)\}$ with a linear combination of D cubic B-spline basis functions $\Psi\{x_p(t)\} = \{\psi_1(x_p(t)), \psi_2(x_p(t)), \dots, \psi_D(x_p(t))\}^\top$, i.e.,

$$\hat{f}_{ip}\{x_p(t)\} = \Psi\{x_p(t)\}^\top \mathbf{a}_{ip}, \quad (3.3)$$

where the column vector $\mathbf{a}_{ip} \in \mathbb{R}^D$ denotes the basis coefficients for component $i, p = 1, \dots, I$. Consequently, model (3.1) is approximated by

$$\hat{x}'_i(t) = a_{i0} + \sum_{p=1}^I \Psi\{x_p(t)\}^\top \mathbf{a}_{ip}, \quad i = 1, 2, \dots, I. \quad (3.4)$$

With the measurement model (1.2) aforementioned, let y_{ij} denote the noisy observation of $x_i(t)$ made at time t_j for $i = 1, \dots, I$ and $j = 1, \dots, J$, i.e., the i th component in $\mathbf{y}_j$. For concreteness in the following exposition, the measurement error ϵ_{ij} is assumed to

be independent Gaussian with variance σ_i^2 , which leads to the following ideal likelihood function of the observations:

$$p(\mathbf{y}|\mathbf{x}(t), \boldsymbol{\sigma}) \propto \left(\prod_{i=1}^I \prod_{j=1}^J \sigma_i^2 \right)^{-1/2} \exp \left(- \sum_{i=1}^I \left[\sum_{j=1}^J \frac{\{y_{ij} - x_i(t_j)\}^2}{2\sigma_i^2} \right] \right). \quad (3.5)$$

As in Ramsay et al. (2007), we approximate $x_i(t)$ using an expansion of L cubic B-spline basis functions $\Phi(t) = (\phi_1(t), \phi_2(t), \dots, \phi_L(t))^\top$, i.e.,

$$\hat{x}_i(t) = \Phi(t)^\top \mathbf{b}_i, \quad (3.6)$$

where the column vector $\mathbf{b}_i \in \mathbb{R}^L$ denotes the basis coefficients for component $i = 1, \dots, I$. Consequently, the likelihood function (3.5) is replaced by the spline-approximated version

$$p(\mathbf{y}|\mathbf{b}, \boldsymbol{\sigma}) \propto \left(\prod_{i=1}^I \prod_{j=1}^J \sigma_i^2 \right)^{-1/2} \exp \left(- \sum_{i=1}^I \left[\sum_{j=1}^J \frac{\{y_{ij} - \Phi(t_{ij})^\top \mathbf{b}_i\}^2}{2\sigma_i^2} \right] \right). \quad (3.7)$$

Note that under this approximation, an estimate of the initial condition of each component is given by $\hat{x}_i(0) = \Phi(0)^\top \mathbf{b}_i$.

Frequentist methods usually assume $\mathbf{b}_i$'s in (3.6) are fixed but unknown parameters (Wu et al., 2014; Chen et al., 2017; Dai and Li, 2022). In contrast, under the Bayesian framework, we need to assign an appropriate prior to facilitate the estimation of $\mathbf{b} = (\mathbf{b}_1^\top, \dots, \mathbf{b}_I^\top)^\top$, for which we leverage the information of model (3.1). When the form of the function $\mathbf{f}$ in ODE system (1.1) is known, one approach is to directly incorporate (1.1) into the prior (Wang et al., 2022). However, this procedure entails an estimation of the derivative $\mathbf{x}'(t)$, which would result in a slower convergence rate than estimating the function $\mathbf{x}(t)$ itself (Stone, 1985; Fan and Gijbels, 1995). Under the linear ODE model assumption, Dattner and Klaassen (2015) bypassed the derivative estimation from the frequentist perspective. In particular, they first integrated both sides of (1.1), and then considered the discrepancy between $\hat{\mathbf{x}}(t)$ and the integral $\int \hat{\mathbf{f}} \{\hat{\mathbf{x}}(t)\} dt$, rather than that between $\hat{\mathbf{x}}'(t)$ and $\hat{\mathbf{f}} \{\hat{\mathbf{x}}(t)\}$. Following this idea, Xu et al. (2024) relaxed this linear ODE assumption to general nonlinear ODEs with known function $\mathbf{f}$ and proposed a Bayesian method that incorporates the integral form of (1.1) into the prior. In the case of an unknown form for $\mathbf{f}$, Chen et al. (2017) extended this integral-based technique to additive sparse ODEs (3.1).

For our Bayesian structure, we consider a similar integral-based prior as in Xu et al. (2024) that uses the spline approximation $\hat{\mathbf{x}}(t)$; but in contrast, the unknown function $\mathbf{f}$ is

now substituted with its spline approximation $\hat{\mathbf{f}}$, resulting in the prior

$$\begin{aligned}\tilde{\pi}_0(\mathbf{b} \mid \mathbf{a}, \lambda) &\propto \exp \left(-\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\hat{x}_i(t) - \int_0^t \hat{f}_i\{\hat{\mathbf{x}}(s)\} ds - \hat{x}_i(0) \right]^2 dt \right) \\ &\propto \exp \left(-\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\Phi(t)^\top \mathbf{b}_i - \int_0^t \left\{ a_{i0} + \sum_{p=1}^I \Psi(\Phi(s)^\top \mathbf{b}_p)^\top \mathbf{a}_{ip} \right\} ds - \Phi(0)^\top \mathbf{b}_i \right]^2 dt \right),\end{aligned}\tag{3.8}$$

where a smoothing parameter $\lambda > 0$ is imposed to control the trade-off between the fit to the data and the fidelity to the ODE system. In particular, a relatively large λ encourages a closer B-spline approximation to the dynamics specified by the ODEs. To estimate and select the non-zero functions among the $\hat{f}_{ip}$'s defined in (3.3), we assign a spike-and-slab group lasso prior (Bai et al., 2022) on $\mathbf{a}_{ip}$, which takes the form

$$\tilde{\pi}_0(\mathbf{a}_{ip} \mid \sigma_i, \theta_{ip}) = \theta_{ip} \delta(\mathbf{a}_{ip} \mid \sigma_i, \eta_0) + (1 - \theta_{ip}) \delta(\mathbf{a}_{ip} \mid \sigma_i, \eta_1), \quad \text{with } \eta_1 \gg \eta_0 > 0,$$

where $\theta_{ip} \in (0, 1)$ is the mixing proportion, and $\delta(\mathbf{a}_{ip} \mid \sigma_i, \eta)$ denotes the Laplace density indexed by hyperparameters σ_i and η :

$$\delta(\mathbf{a}_{ip} \mid \sigma_i, \eta) \propto \left(\frac{\eta}{\sigma_i} \right)^D \exp \left(-\frac{\eta}{\sigma_i} \|\mathbf{a}_{ip}\|_2 \right),\tag{3.9}$$

where η/σ_i describes the rate of $\mathbf{a}_{ip}$ approaching zero. When η/σ_i is relatively large, the prior distribution would be concentrated at zero, shrinking $\mathbf{a}_{ip}$ towards zero. In contrast, when η/σ_i is small, $\mathbf{a}_{ip}$ can deviate from zero with a high probability. To select an appropriate θ_{ip} when estimating $\mathbf{a}_{ip}$, we assign a Beta prior on θ_{ip} (Bai et al., 2022; Alhamzawi and Ali, 2018),

$$\tilde{\pi}_0(\theta_{ip} \mid \alpha_\theta, \beta_\theta) \propto \theta_{ip}^{\alpha_\theta - 1} (1 - \theta_{ip})^{\beta_\theta - 1},\tag{3.10}$$

where $\alpha_\theta > 0$ and $\beta_\theta > 0$ are the shape parameters. Without prior knowledge, Alhamzawi and Ali (2018) recommended setting $\alpha_\theta = \beta_\theta = 1$, which leads to a uniform prior on $[0, 1]$. For the remaining parameters, we may assign the generic non-informative priors $\tilde{\pi}_0(a_{i0}) \propto 1$ and $\tilde{\pi}_0(\sigma_i) \propto 1/\sigma_i$ independently. Then the posterior distribution of $(\mathbf{a}, \mathbf{b}, \boldsymbol{\sigma}, \boldsymbol{\theta})$ becomes

$$\begin{aligned}\pi(\mathbf{a}, \mathbf{b}, \boldsymbol{\sigma}, \boldsymbol{\theta} \mid \mathbf{y}, \lambda) &\propto \pi(\mathbf{y} \mid \mathbf{a}, \mathbf{b}, \boldsymbol{\sigma}, \boldsymbol{\theta}, \lambda) \pi(\mathbf{b} \mid \mathbf{a}, \boldsymbol{\sigma}, \boldsymbol{\eta}, \lambda) \tilde{\pi}_0(\mathbf{a} \mid \boldsymbol{\sigma}, \boldsymbol{\theta}) \tilde{\pi}_0(\boldsymbol{\sigma}) \tilde{\pi}_0(\boldsymbol{\theta}) \\ &= p(\mathbf{y} \mid \mathbf{b}, \boldsymbol{\sigma}) \tilde{\pi}_0(\mathbf{b} \mid \mathbf{a}, \lambda) \tilde{\pi}_0(\mathbf{a} \mid \boldsymbol{\sigma}, \boldsymbol{\theta}) \tilde{\pi}_0(\boldsymbol{\sigma}) \tilde{\pi}_0(\boldsymbol{\theta}),\end{aligned}\tag{3.11}$$

as in the factorization, $\mathbf{y}$ only depends on $\mathbf{b}$ and $\boldsymbol{\sigma}$ through (3.7), and $\mathbf{b}$ only depends on $\mathbf{a}$ through (3.8).

3.2.2 Integral Estimation

Given that the posterior distribution (3.11) contains two integrals which usually do not have closed-form expressions, numerical techniques are entailed to approximate them. We apply the Gaussian quadrature rule (Stroud and Secrest, 1966) to these two integrals. Specifically, for the outer integral, the expansion is replaced by

$$\begin{aligned} \mathbf{U}_i &= \int_{t_1}^{t_J} \left[\Phi(t)^\top \mathbf{b}_i - \int_0^t \left\{ a_{i0} + \sum_{p=1}^I \Psi(\Phi(s)^\top \mathbf{b}_p)^\top \mathbf{a}_{ip} \right\} ds - \Phi(0)^\top \mathbf{b}_i \right]^2 dt \\ &\approx \sum_{m=1}^{M_i} v_m \left[\Phi(\xi_m)^\top \mathbf{b}_i - \int_0^{\xi_m} \left\{ a_{i0} + \sum_{p=1}^I \Psi(\Phi(s)^\top \mathbf{b}_p)^\top \mathbf{a}_{ip} \right\} ds - \Phi(0)^\top \mathbf{b}_i \right]^2, \end{aligned} \quad (3.12)$$

where M_i denotes the number of quadrature points used to evaluate the outer integral, $\{\xi_m\}_{m=1}^{M_i}$ are the quadrature points inside the interval $[t_1, t_J]$, and $\{v_m\}_{m=1}^{M_i}$ are the corresponding quadrature weights. For the inner integral, the expansion is expressed as

$$\begin{aligned} \mathbf{O}_{m_i} &= \int_0^{\xi_m} \left\{ a_{i0} + \sum_{p=1}^I \Psi(\Phi(s)^\top \mathbf{b}_p)^\top \mathbf{a}_{ip} \right\} ds \\ &= a_{i0} \xi_m + \sum_{p=1}^I \left[\int_0^{\xi_m} \Psi\{\Phi(s)^\top \mathbf{b}_p\} ds \right]^\top \mathbf{a}_{ip} \\ &\approx a_{i0} \xi_m + \sum_{p=1}^I \left[\sum_{k=1}^{K_{m_i}} w_k \Psi\{\Phi(s_k)^\top \mathbf{b}_p\} \right]^\top \mathbf{a}_{ip}, \end{aligned} \quad (3.13)$$

where K_{m_i} is the number of quadrature points used to evaluate the inner integral, $\{s_k\}_{k=1}^{K_{m_i}}$ are the quadrature points inside the interval $[0, \xi_m]$, and $\{w_k\}_{k=1}^{K_{m_i}}$ are the corresponding quadrature weights.

According to the guidelines in Xu et al. (2024), we use roughly $(M+1)(L-2)/2$ quadrature points for the outer integral (3.12), where M and $(L-2)$ denote the degree and the number of interior knots of the B-spline Φ , respectively. For the inner integral,

we use the exact 2 quadrature points for the cubic B-spline Ψ as in [de Boor et al. \(1976\)](#). More details about the choice for the number of quadrature points can be found in [Section C.1](#) of the appendix.

3.2.3 Identifiability

As stated in [Section 3.2.1](#), the identifiability of the additive components f_{ip} can be assured by the constraint [\(3.2\)](#). With the B-spline approximation of function f_{ip} in [\(3.3\)](#), this constraint is equivalent to centering the basis function Ψ and spline approximation $\hat{x}(t)$ with mean zero in [\(3.12\)](#) and [\(3.13\)](#), i.e., we replace $\mathbf{U}_i$ with

$$\tilde{\mathbf{U}}_i = \sum_{m=1}^{M_i} v_m \left[\tilde{\Phi}(\xi_m)^\top \mathbf{b}_i - a_{i0} \tilde{\xi}_m - \sum_{p=1}^I \left\{ \sum_{k=1}^{K_{m_i}} w_k \tilde{\Psi}(\Phi(s_k)^\top \mathbf{b}_p) \right\}^\top \mathbf{a}_{ip} - \tilde{\Phi}(0)^\top \mathbf{b}_i \right]^2, \quad (3.14)$$

where

$$\begin{aligned} \tilde{\Phi}(\xi_m) &= \Phi(\xi_m) - M_i^{-1} \sum_{m=1}^{M_i} \Phi(\xi_m), \quad \tilde{\xi}_m = \xi_m - M_i^{-1} \sum_{m=1}^{M_i} \xi_m, \\ \text{and } \tilde{\Psi}\{\Phi(s_k)^\top \mathbf{b}_p\} &= \Psi\{\Phi(s_k)^\top \mathbf{b}_p\} - (K_{m_i} M_i)^{-1} \sum_{m=1}^{M_i} \sum_{k=1}^{K_{m_i}} \Psi\{\Phi(s_k)^\top \mathbf{b}_p\}. \end{aligned}$$

3.2.4 Practical Implementation

The posterior distribution [\(3.11\)](#) is implemented in C++ and passed to *tmbstan* ([Monnahan and Kristensen, 2018](#)). NUTS ([Homan and Gelman, 2014](#)) serves as the sampling algorithm, which enables a more efficient exploration of the posterior distribution compared to traditional MCMC samplers. One chain of 1000 iterations, including a 500-iteration warmup, is run for our algorithm in all of the subsequent examples. All these experiments are performed on two 8-core Xeon processors at 3.6 GHz and 256 GB RAM memory. More details on the implementation can be found in [Section A](#) of the appendix.

To set an appropriate value for η_0 and η_1 , note that an excessively large η_1 will introduce significant bias into the resulting estimator ([Bai, 2020](#)), while the value of η_0 should be small enough to ensure that the slab is sufficiently diffuse ([Bai et al., 2021](#)). As a result,

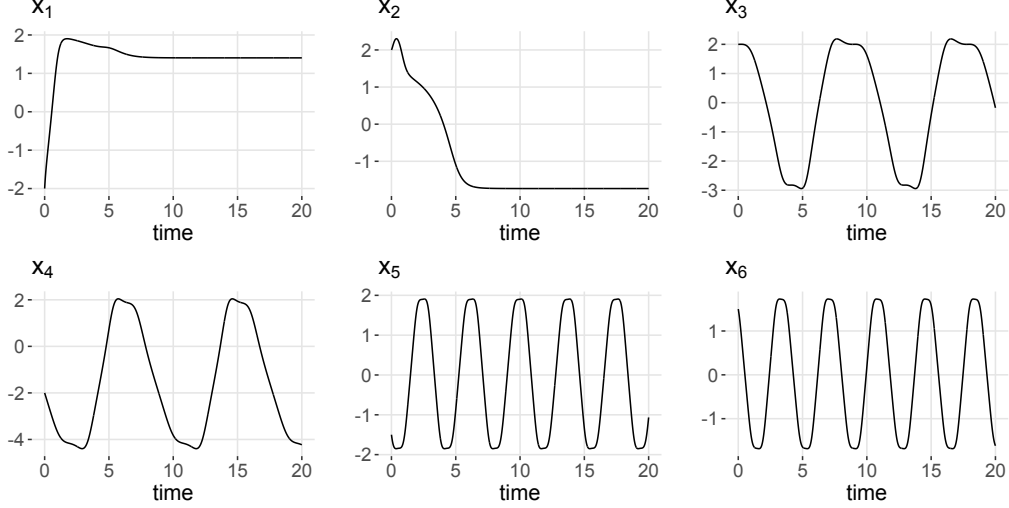


Figure 3.1: The true trajectories $\mathbf{x}_1, \dots, \mathbf{x}_6$ of the simulated ODE system (3.15) in the time interval $[0, 20]$.

we recommend setting the value of η_0 such that $\eta_0/\hat{\sigma}_i$ is roughly equal to or greater than 1, and the value of η_1 such that $\eta_1/\hat{\sigma}_i$ is approximately greater than 20 but less than 100.

To obtain the initial estimates, we employ the R package *CollocInfer* (Hooker et al., 2016) to smooth the data with the roughness penalty 0.01 and treat the coefficients as $\hat{\mathbf{b}}^{(0)}$. For initializing $\hat{\mathbf{a}}^{(0)}$, the group lasso regression in the package *grpreg* (Breheny, 2015) is used with the response variable set as $\hat{\mathbf{x}}_i^{(0)} = \Phi^\top \hat{\mathbf{b}}_i^{(0)}$. Based on this initialization, we set $\hat{\theta}_{ip}^{(0)} = 1$ if $\hat{\mathbf{a}}_{ip}^{(0)} = \mathbf{0}$, and $\hat{\theta}_{ip}^{(0)} = 0$ if $\hat{\mathbf{a}}_{ip}^{(0)} \neq \mathbf{0}$. However, it should be noted that the Laplace density (3.9) for $\mathbf{a}$ is not differentiable at zero, so any $\hat{\mathbf{a}}_{ip}^{(0)} = \mathbf{0}$ cannot be used directly to initialize NUTS; hence, we add a small $\text{Normal}(0, 10^{-10})$ random jitter to zero values in $\hat{\mathbf{a}}^{(0)}$. Finally, we set $\hat{\sigma}_i^{(0)} = 0.1$.

3.3 Simulation Study

3.3.1 Benchmark System and Setup

In this simulation study, we evaluate the performance of our method based on the examples from Zhang et al. (2022) and Chen et al. (2017) with the number of variables $I = 10$. In

particular, the ODE system takes the form

$$\begin{cases} x'_{2k-1}(t) = \beta_{2k-1,0} + \boldsymbol{\beta}_{2k-1,1}^\top \mathbf{h}\{x_{2k-1}(t)\} + \boldsymbol{\beta}_{2k-1,2}^\top \mathbf{h}\{x_{2k}(t)\} \\ x'_{2k}(t) = \beta_{2k,0} + \boldsymbol{\beta}_{2k,1}^\top \mathbf{h}\{x_{2k-1}(t)\} + \boldsymbol{\beta}_{2k,2}^\top \mathbf{h}\{x_{2k}(t)\} \end{cases}, \quad (3.15)$$

where $k = 1, \dots, 5$, and $\mathbf{h}(x) = (x, x^2, x^3)^\top$ denotes the cubic monomial basis. As in [Zhang et al. \(2022\)](#), the initial values and parameters are specified as follows:

1. $(x_1(t), x_2(t))^\top$ is generated with $\beta_{1,0} = 0$, $\boldsymbol{\beta}_{1,1} = (1.2, 0.3, -0.6)^\top$, $\boldsymbol{\beta}_{1,2} = (0.1, 0.2, 0.2)^\top$, $\beta_{2,0} = 0.4$, $\boldsymbol{\beta}_{2,1} = (-2, 0, 0.4)^\top$, $\boldsymbol{\beta}_{2,2} = (0.5, 0.2, -0.3)^\top$, and initial values $x_1(0) = -2$, $x_2(0) = 2$.
2. $(x_3(t), x_4(t))^\top$ is generated with $\beta_{3,0} = -0.2$, $\boldsymbol{\beta}_{3,1} = (0, 0, 0)^\top$, $\boldsymbol{\beta}_{3,2} = (0.3, 0.4, 0.1)^\top$, $\beta_{4,0} = -0.2$, $\boldsymbol{\beta}_{4,1} = (0.2, 0.1, -0.2)^\top$, $\boldsymbol{\beta}_{4,2} = (0, 0, 0)^\top$, and initial values $x_3(0) = 2$, $x_4(0) = -2$.
3. $(x_5(t), x_6(t))^\top$ is generated with $\beta_{5,0} = 0.05$, $\boldsymbol{\beta}_{5,1} = (0, 0, 0)^\top$, $\boldsymbol{\beta}_{5,2} = (0.1, 0, -0.8)^\top$, $\beta_{6,0} = -0.05$, $\boldsymbol{\beta}_{6,1} = (0, 0, 0.5)^\top$, $\boldsymbol{\beta}_{6,2} = (0, 0, 0)^\top$, and initial values $x_5(0) = -1.5$, $x_6(0) = 1.5$.
4. $(x_7(t), x_8(t), x_9(t), x_{10}(t))^\top$ is generated with $\beta_{7,0}, \beta_{8,0}, \beta_{9,0}, \beta_{10,0}$ and initial values drawn from the standard normal distribution, and all other parameters set to zero. In other words, these four variables $x_i(t), i = 7, \dots, 10$ are linear in t .

Therefore, this ODE system can be expressed in the form of (3.1), with only eight out of one hundred functions $f_{ip}, i, p = 1, \dots, 10$ being non-zero. Following the same setup as in [Zhang et al. \(2022\)](#), we generate 41 equally-spaced noisy observations of $\mathbf{x}_i, i = 1, \dots, 10$ in the time interval $[0, 20]$, and the true trajectories $\mathbf{x}_1, \dots, \mathbf{x}_6$ are displayed in Figure 3.1. Additionally, we consider the case with 21 noisy observations to demonstrate the effectiveness of our method with sparser observations. To explore the performance of our method at different noise levels, we choose the signal-to-noise ratio (SNR), which is defined here as the ratio between the sample standard deviation of $\{\mathbf{x}_i(t_j)\}_{j=1}^J$ and the standard deviation σ_i ([Nanshan et al., 2022](#)), to be 25, 10 and 4, respectively. It should be noted that the lower the SNR, the higher the noise level. For all SNR levels, we place 41 and 21 knots at exactly each observed time point for $\Phi(t)$, under the case of 41 and 21 observations, respectively. One interior knot is placed on $\hat{\mathbf{x}}(t)$ without an intercept for each basis function $\Psi(\hat{\mathbf{x}}(t))$. Following the guideline in Section 3.2.2, $M_i = 100$ and $M_i = 50$ quadrature points are chosen to estimate the outer integral (3.14) for 41 and 21 observations, respectively. Since the observations are relatively dense in the time interval,

we fix $\lambda = 1000$ according to the simulation results in [Qi and Zhao \(2010\)](#) and [Ramsay et al. \(2007\)](#). For the hyperparameters in the prior of θ in (3.10), we set $\alpha_\theta = \beta_\theta = 1$ as recommended by [Alhamzawi and Ali \(2018\)](#). Finally, we set $\eta_1 = 10$ and $\eta_0 = 0.25$.

To evaluate the performance of our method, we compare it with JADE ([Zhang et al., 2022](#)) and GRADE ([Chen et al., 2017](#)). We strictly follow the recommendations of the authors: for JADE, a maximum of four iterations are run in their block descent algorithm to update the parameters, and a cubic spline basis with four knots is chosen for the spline approximation of f_{ip} ; for GRADE, we use the default configurations¹². We run 100 independent simulation trials to evaluate the estimation performance. Both comparison methods take the sample means as the parameter estimates, while our method takes the posterior means of the sample. In all simulation studies, the R package *deSolve* ([Soetaert et al., 2010](#)) is used for numerically solving the differential equations to generate the simulation data.

3.3.2 Performance Assessment

We consider assessing the performance of different methods from two aspects: the additive ODE component estimates $\hat{f}_{ip}$ and the estimated trajectories $\hat{\mathbf{x}}$. Regarding the trajectory estimation, we consider the trajectory mean squared error (MSE), which is defined as

$$\text{MSE}(\hat{\mathbf{x}}) = \frac{1}{I} \sum_{i=1}^I \int_{t_1}^{t_J} \{\hat{x}_i(t) - x_i(t)\}^2 dt,$$

where $\hat{x}_i(t)$ and $x_i(t)$ denote the estimated and true trajectories, respectively. Notice that these two trajectories are obtained from different methods: for the estimated trajectories, we consider the spline approximation (3.6); for the true trajectories, we apply the numerical solver to the simulation ODE system (3.15). For the additive ODE components, following [Zhang et al. \(2022\)](#), the MSEs for the non-zero and zero ones are calculated separately,

¹²More details for the implementation of GRADE for this simulation study can be found in Section 5.1 of [Chen et al. \(2017\)](#).

i.e.,

$$\begin{aligned}\text{MSE}_{\mathcal{A}}(\hat{f}) &= \frac{1}{|\mathcal{A}|} \sum_{f_{ip} \in \mathcal{A}} \int_{t_1}^{t_J} \left\{ \hat{f}_{ip}(\mathbf{x}_p) - f_{ip}(\mathbf{x}_p) \right\}^2 d\mathbf{x}_p, \\ \text{MSE}_{\mathcal{A}^c}(\hat{f}) &= \frac{1}{|\mathcal{A}^c|} \sum_{f_{ip} \in \mathcal{A}^c} \int_{t_1}^{t_J} \left\{ \hat{f}_{ip}(\mathbf{x}_p) - f_{ip}(\mathbf{x}_p) \right\}^2 d\mathbf{x}_p,\end{aligned}$$

where $\mathcal{A} = \{f_{ip} : f_{ip} \neq 0\}$, $\mathcal{A}^c = \{f_{ip} : f_{ip} = 0\}$, and $|\cdot|$ denotes the cardinality of a set. We also consider the MSEs for all f_{ip} 's, i.e., $f_{ip} \in \mathcal{A} \cup \mathcal{A}^c$. Moreover, for evaluating the performance of system recovery, the true-positive rate (TPR) and false-positive rate (FPR) are calculated to assess the accuracy of identifying non-zero f_{ip} , where we treat $\hat{f}_{ip}$ as non-zero if $\hat{\theta}_{ip} > 0.5$ for our method and $\hat{\mathbf{a}}_{ip} = 0$ for frequentist methods.

3.3.3 Results

Table 3.1: Comparison of methods for 41 observations in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.

SNR	Method	MSE($\hat{\mathbf{x}}$) ($\times 10^{-2}$)	MSE $_{\mathcal{A}}(\hat{f})$	MSE $_{\mathcal{A}^c}(\hat{f})$ ($\times 10^{-2}$)	MSE($\hat{f}$)	TPR	FPR
25	Integral	0.190	0.081	0.003	0.007	0.999	0.167
		(0.020)	(0.006)	(0.008)	(0.001)	(0.013)	(0.043)
	JADE	0.136	0.113	0.186	0.010	0.895	0.052
		(0.018)	(0.055)	(0.538)	(0.006)	(0.090)	(0.035)
	GRADE	0.268	0.147	0.166	0.013	1.000	0.109
		(0.027)	(0.021)	(0.301)	(0.003)	(0.000)	(0.048)
10	Integral	0.514	0.115	0.029	0.010	0.968	0.059
		(0.076)	(0.023)	(0.076)	(0.002)	(0.047)	(0.030)
	JADE	0.602	0.151	0.769	0.019	0.856	0.098
		(0.120)	(0.052)	(1.335)	(0.015)	(0.146)	(0.067)
	GRADE	1.119	0.272	0.444	0.026	0.979	0.066
		(0.183)	(0.035)	(1.698)	(0.016)	(0.047)	(0.034)
4	Integral	2.436	0.253	0.277	0.023	0.768	0.055
		(0.484)	(0.048)	(0.305)	(0.006)	(0.107)	(0.030)
	JADE	3.486	0.392	4.842	0.076	0.848	0.203
		(1.498)	(0.232)	(4.562)	(0.052)	(0.160)	(0.103)
	GRADE	4.715	0.510	0.386	0.044	0.864	0.039
		(1.040)	(0.106)	(1.036)	(0.017)	(0.098)	(0.025)

Table 3.1 summarizes the results of the three methods for 41 observations under different noise levels, where the mean and standard error of each performance metric are presented. Concerning the trajectory recovery, since the observations are relatively dense on the time grid, all methods estimate the trajectories reasonably well. Particularly, our method has a slightly higher trajectory MSE than JADE when the SNR is 25; when the SNR decreases to 10, our method has a lower trajectory MSE than JADE. This difference becomes more evident as the SNR decreases to 4. Under all SNR levels, GRADE has the highest trajectory MSE among all methods. In terms of estimating the non-zero ODE additive components, our proposed integral method outperforms the competitors under all noise levels. Re-

garding the MSE for estimating vanishing components, our method has a clear advantage compared to other methods. GRADE has an overall lower $\text{MSE}_{\mathcal{A}^c}(\hat{f})$ than JADE with fewer false positives, which is in line with the findings in [Chen et al. \(2017\)](#). Furthermore, for both recovering the trajectories and estimating the additive components, our method has a lower standard error for each performance metric compared with other methods, which demonstrates its stability. For network discovery, all methods can identify at least around 85% of the true non-zero additive components when the SNR is 25 or 10. Despite that our method has a slightly lower TPR than GRADE, it can provide the best estimation for the non-zero components compared to all other methods. For identifying the vanishing components, when the noise level increases, the FPR of our method and GRADE will decrease, while JADE tends to have a higher FPR. While GRADE has the best TPR for identifying non-zero additive components among all methods, its estimation error is the highest across all noise levels. Thus, we can conclude that both TPR and FPR might be of limited importance, as our method achieves the lowest estimation error for additive components across all noise levels. Nevertheless, our method still provides competitive performance in network discovery, in terms of a relatively high TPR and a fairly low FPR.

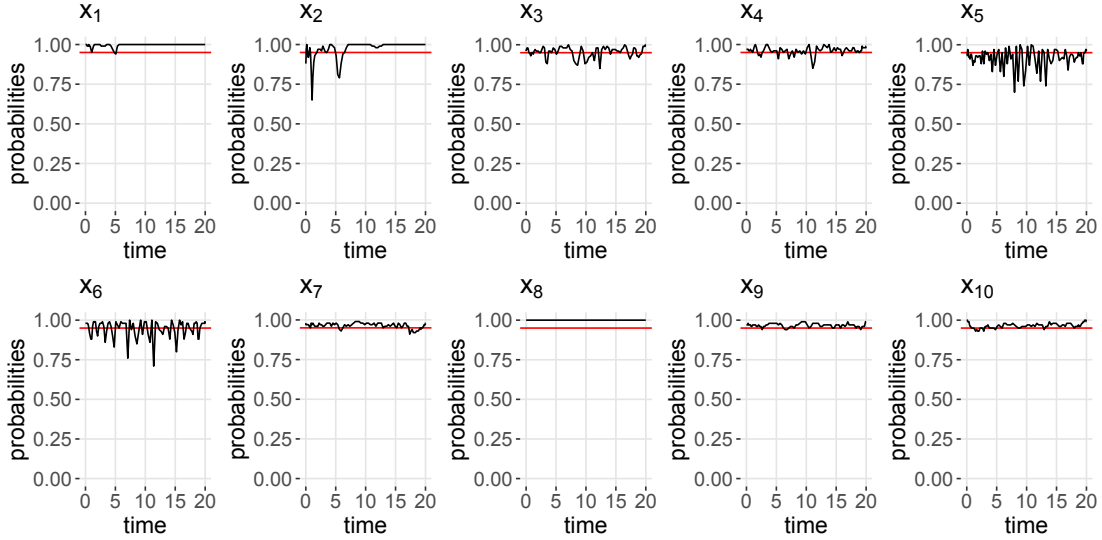


Figure 3.2: The pointwise coverage probabilities of each trajectory in the simulated ODE system for 41 observations over 100 simulation runs when SNR=10 (black line). The red horizontal line indicates the nominal level 95%.

To showcase the performance of our method in estimating the ODE solution under the additive assumption, we consider the estimated integral form of (3.4), i.e., $\hat{x}_i(t) =$

$\hat{a}_{i0}t + \int_{t_1}^{t_J} \sum_{p=1}^I \Psi(\hat{x}_p(s))^\top \hat{\mathbf{a}}_{ip} ds$. The pointwise coverage probabilities are calculated to quantify the estimation uncertainty: if the true $x_i(t)$ is contained within the 95% central interval of the posterior samples for the estimated $\hat{x}_i(t)$, then we treat it as covered at point t . From the frequentist perspective, Dai and Li (2022) argued that the actual coverage probabilities can be dramatically lower than the nominal level if ignoring the uncertainty induced by the sparse estimation. Therefore, they constructed universally valid post-selection confidence intervals (Berk et al., 2013) based on estimated trajectories $\hat{\mathbf{x}}(t)$ under the kernel ODE framework. Dai and Li (2024) further introduced a local approximation technique under the same framework to construct post-regularization confidence bands for ODEs using a de-biasing method. Nevertheless, there is no straightforward or standard implementation of their procedure to construct such confidence intervals. Figure 3.2 shows the pointwise coverage probabilities of our proposed method in the simulated ODE system across 100 simulation runs for SNR = 10, while those for SNR = 25 and 4 are shown in Figures C.1 and C.2 of the appendix, respectively. As the 95% central interval of $\hat{x}_i(t)$ becomes wider under the scenario of higher noise levels, we may expect higher overall pointwise coverage probabilities, which are closer to the nominal level 95%, especially for trajectories $\mathbf{x}_2, \mathbf{x}_5$ and $\mathbf{x}_6$. On the other hand, as the noise level increases, the estimation for the trajectories can be less precise, particularly for trajectories $\mathbf{x}_7$ to $\mathbf{x}_{10}$ which are linear in time. As a result, the coverage probabilities for these trajectories decrease from nearly 100% to around the nominal level. It should be noted that when noise levels are exceptionally high, the coverage probabilities might drop below the nominal level, especially for those nonlinear trajectories as depicted in Figure C.2. Nevertheless, our method can provide reliable inference for the ODE solution for all noise levels, where the overall pointwise coverage probabilities for trajectories are close to the nominal level.

Similar to Table 3.1, the results for 21 observations with different noise settings are displayed in Table C.1 of the appendix. In terms of recovering trajectories, our method has a slightly higher trajectory MSE than other methods when the SNR is 25 or 10; however, the trajectory MSE of our method is the lowest among all methods when the SNR decreases to 4. Similar to the case of 41 observations, our method still achieves the lowest MSEs among all methods across all noise levels for both zero and non-zero additive components, while GRADE has the highest MSEs. Furthermore, as the noise level increases, this advantage of our method becomes more apparent. Regarding network recovery, although GRADE has a dominating performance for TPR for each noise level, it has a considerably higher FPR compared with other methods. Moreover, our method can select true non-zero additive components more frequently than JADE. Figures C.3–C.5 in the appendix present the pointwise coverage probabilities under different noise levels with 21 observations, with conclusions similar to our findings for the case with 41 noisy

observations. Further discussion is provided in Section C.2.1 in the appendix.

We also examine the applicability of our method to larger systems by considering the simulation system (3.15) with $I = 16$. The initial values and parameters of x_i , for $i = 1, \dots, 10$, are the same as in Section 3.1, while the new variables x_i , for $i = 11, \dots, 16$, are all linear in time t . Furthermore, we investigate the robustness of our method to violations of the additive ODE and B-spline approximation assumptions through two additional simulation studies, which consider a system with interactions and a system with absolute value functions, respectively. The overall conclusions of these three simulation studies are consistent with our findings for the simulation system (3.15). Specifically, the computation time of our method grows approximately linearly with the number of variables I , and the inference remains reliable. Even when the additive ODE or B-spline approximation assumptions are violated, our method may still adequately recover the system. More details are provided in Section C.2 of the appendix.

3.4 Real Data Analysis

As a critical part of systems biology, the associations among genes and their products are studied to shed light on the molecular mechanisms of cellular function (Hausser and Strimmer, 2009). Extensive models have been developed for identifying the structure of gene regulatory networks (GRNs) from noisy time-course gene expression data. Specifically, if f_i is dependent on x_p , then we say x_p regulates x_i , i.e., x_p controls the change of x_i through the functional f_i on the derivative dx_i/dt . In other words, we are interested in identifying which gene regulates another. In this study, we focus on the identification of a nonlinear dynamic GRN based on the investigation by Rangel et al. (2005) for the response of human T-cells to phorbol myristyl acetate (PMA) and ionomycin treatment. For the 58 T-cell activation genes monitored by an experiment in Rangel et al. (2004), gene expression levels were measured at 10 unequally spaced time points (0, 2, 4, 6, 8, 18, 24, 32, 48, 72 hour) for 34 replicates. More details about data acquisition can be found in Rangel et al. (2005).

For implementing our method, we place 10 interior knots corresponding to each observed time point for $\Phi(t)$, and place 2 interior knots on $\hat{\mathbf{x}}(t)$ (without intercept) for each basis function $\Psi(\hat{\mathbf{x}}(t))$. Following the recommendation in Alhamzawi and Ali (2018), we set the hyperparameters $\alpha_\theta = \beta_\theta = 1$ for the prior of θ in (3.10). According to the guideline in Section 3.2.2, $M_i = 30$ quadrature points are taken to estimate the outer integral (3.14). Since observations are relatively sparse over the time interval, we fix $\lambda = 10$ in our algorithm, following the simulation and real data results from Xu et al. (2024), to better

balance the fit to the data and ODE fidelity. Finally, we choose $\eta_1 = 10$ and $\eta_0 = 0.6$. The other method settings are the same as those used in the simulation studies.

Figure 3.3 depicts the estimated trajectories for four genes and their central 95% point-wise credible intervals based on our method. Given the relatively noisy gene expression measurements at each time point, it makes sense that the 95% credible intervals for the trajectory do not cover all the observations. Furthermore, since the expression levels were measured at unequally spaced time points, the 95% credible interval becomes wider during the time interval between two sparser points to reflect greater uncertainty. Overall, our method provides reasonable fits to the gene expression levels. Figure 3.4 visualizes the variable selection results for the T-cell GRN, while Table C.5 in the appendix details the inward and outward influence genes identified for each gene by our method. Among the 3364 possible functions, only 273 are selected to be non-zero, and each gene is only regulated by a few other genes, demonstrating the sparseness of the network connections. In accordance with the results in Rangel et al. (2004) and Wu et al. (2014), FYB (gene 45) is identified as one of the crucial genes with the highest number of outward connections. Notably, in addition to five FYB-regulated genes previously found by Rangel et al. (2004) or Wu et al. (2014), we identify five new outward influence genes of FYB: SIVA (gene 6), IL-16 (gene 23), MyD88 (gene 37), CYP19 (gene 50) and caspase 7 (gene 52). Due to the limited number of genes in this T-cell GRN, some regulations identified by our method might be mediated by other genes not included in this network. For instance, gene MyD88 is directly downstream of FYB in our model. This finding can be supported by the findings that Src kinase-associated phosphoprotein 2 (SKAP2) directly interacts with FYB (Liu et al., 1998), and SKAP2 physically associates with MyD88 (Takagane et al., 2022). As another example, our model predicts that FYB directly regulates the expression of IL-16. This regulation can be supported through two mediated pathways: FYB stimulates lymphokine production in T-cells through a pathway involving SLP-76 (Raab et al., 1999), which is known to correlate with CD4+ T-cells (Wang and Peng, 2021); co-culturing chronic lymphocytic leukemia cells with autologous CD4+ T-cells has been shown to enhance IL-16 release (Wu et al., 2023). In summary, numerous regulations identified by our model can be endorsed by the existing scientific literature. Furthermore, there may be connections that have yet to be discovered by scientists. The estimated regulation functions of genes influenced by FYB are presented in Figure 3.5, all demonstrating strong nonlinear regulation effects in relation to FYB expression levels. For example, in the case of caspase 7 (gene 52), the regulatory effect increases at a low FYB expression level and shifts to a negative effect as FYB expression level rises. Consequently, our method effectively identifies nonlinear¹³ regulation effects and captures the dynamic changes of the

¹³In practice, the regulation effects can be simpler (e.g., monotonic or saturating). A simpler set of basis

effects associated with varying gene expression levels.

We also assess sensitivity to prior choices by setting different hyperparameter values of $\boldsymbol{\eta}$ while keeping the ratio η_1/η_0 fixed. The selection of non-zero functions and the computing time indicate that our method is not particularly sensitive to the choice of prior values. Further details are provided in Section C.3 of the appendix.

3.5 Conclusion and Discussion

In this paper, we develop a Bayesian collocation approach with a group-wise sparse penalty to identify generalized sparse additive ODE structures. In particular, our method applies integral estimation rather than derivative estimation. In contrast to previous frequentist methods, we adopt a Bayesian framework, which provides better uncertainty quantification for the parameters and trajectories. Simulation studies demonstrate the favorable performance of the proposed Bayesian integral-based method compared to existing frequentist methods, including the joint derivative-based estimation method and the two-step integral-based method, in terms of identifying and estimating the additive ODE components. Regarding statistical inference, our method can provide reliable pointwise coverage probabilities for ODE trajectories. Furthermore, we illustrate the applicability of our method by constructing a nonlinear dynamic GRN from noisy T-cell activation gene expression data. Our method not only identifies nonlinear regulation effects, but also captures how these effects vary with regulator expression levels.

The simulation studies also examine the stability of our method under various noise levels and sampling frequencies. For the estimation of ODE trajectories and additive components, the performances of all methods worsen when the observations become sparser or noisier. However, the proposed method consistently outperforms the competitors in additive component estimation and with the lowest variability. This advantage becomes more pronounced as the observations become more noisy or more sparse.

We briefly describe an interesting direction for future work. Rather than first-order ODEs (1.1), dynamic behaviors in scientific applications might be considered to be characterized by higher-order ones $\mathcal{D}_i^K x_i(t) = f_i(\mathbf{x}(t), t; \mathbf{a}), t \in [t_1 = 0, t_J], i = 1, \dots, I$, where $\mathcal{D}^K$ is an order- K differential operator and $\mathbf{a}$ denotes the unknown parameter. Suppose that $\mathcal{D}_i^K$ denotes a linear differential operator of order K , i.e., $\mathcal{D}_i^K = D^K + \sum_{k=0}^{K-1} \omega_{ik} D^k$, where $\boldsymbol{\omega} := \{\omega_{ik}\}_{i \leq I, k \leq K}$ are the coefficients of $\mathcal{D}_i^K$'s and D^k is the differential operation d_k/d_t^k . To generalize our method to such higher-order ODEs, an adaptation of Green's

functions can then be chosen based on the data, but we do not consider that setting here.

matching (Tan et al., 2024) to our Bayesian framework could be considered. Specifically, similar to (3.8) as implemented in Section 2.1, we might assign the prior on $\mathbf{b}$ for ODEs of order- K as

$$\tilde{\pi}_0(\mathbf{b} \mid \mathbf{a}, \mathbf{c}, \boldsymbol{\omega}, \lambda) \propto \exp \left(-\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} [\mathcal{T}(\hat{\mathbf{x}} \mid \mathbf{a}, \mathbf{c}, \boldsymbol{\omega})]^2 w(t) dt \right), \quad (3.16)$$

where $\mathcal{T}(\cdot)$ is called the matching equation, $w(t)$ represents a known function on $[t_1, t_J]$ to assign weights to $\mathcal{T}(\cdot)$ and $\mathbf{c} = (\mathbf{c}_i^\top, \dots, \mathbf{c}_I^\top)^\top$ is the unknown parameter in $\mathcal{T}(\cdot)$. Tan et al. (2024) referred to the fitting procedure in (3.16) as Green's matching, and they demonstrated that $\mathbf{c}$ and $\boldsymbol{\omega}$ can be uniquely determined in their estimation approach. From a Bayesian perspective, estimation of $\mathbf{c}$ and $\boldsymbol{\omega}$ might be regularized by appropriately chosen priors, which we leave as future research.

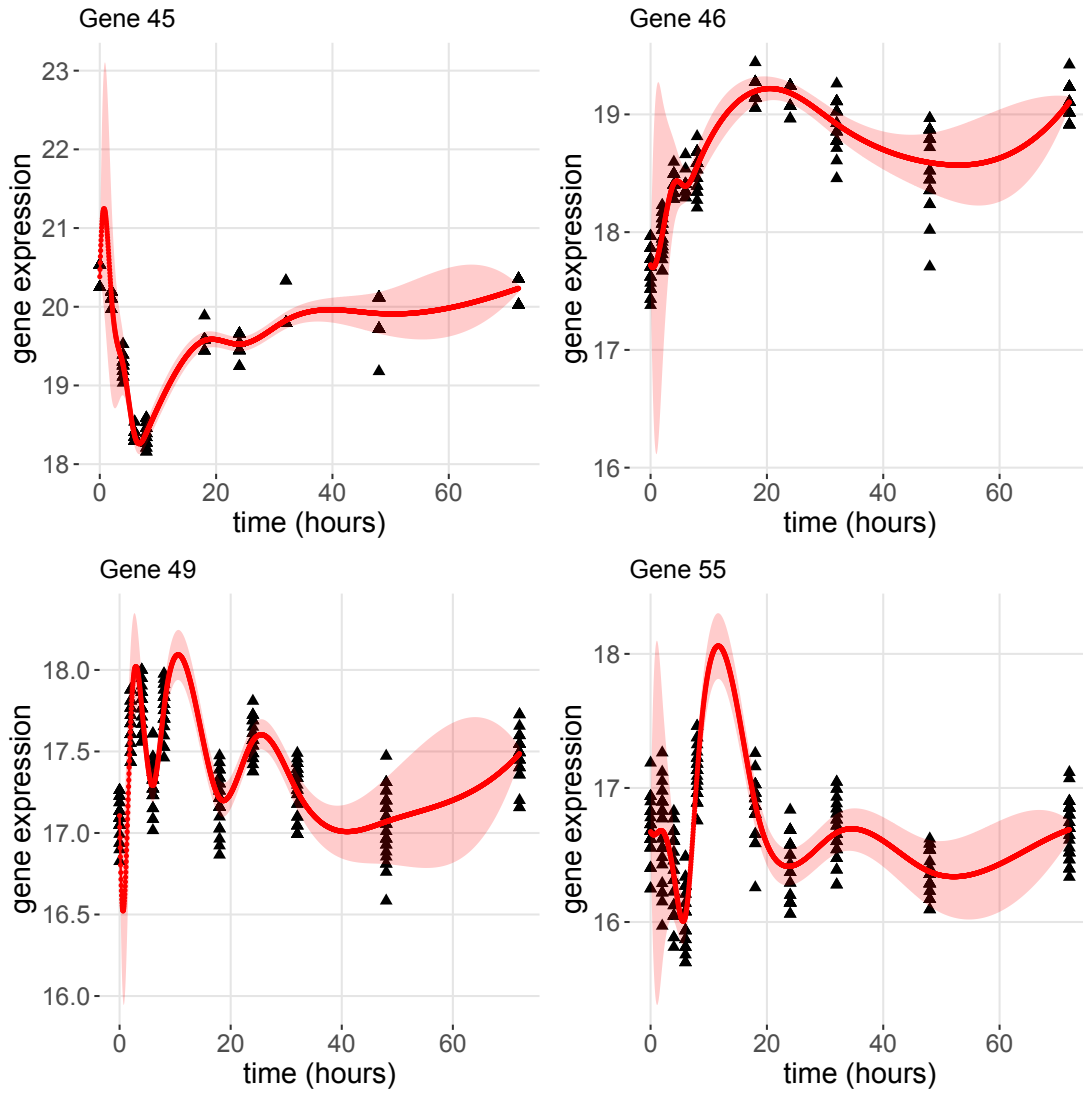


Figure 3.3: Estimated trajectories (gene expression levels) for gene 45 (FYB), 46 (IL-2R γ), 49 (apoptosis) and 55 (IL3R α). The black triangles represent the real data and the red solid lines are the estimated trajectories. The red shaded areas are central 95% pointwise credible intervals for the estimated trajectories.

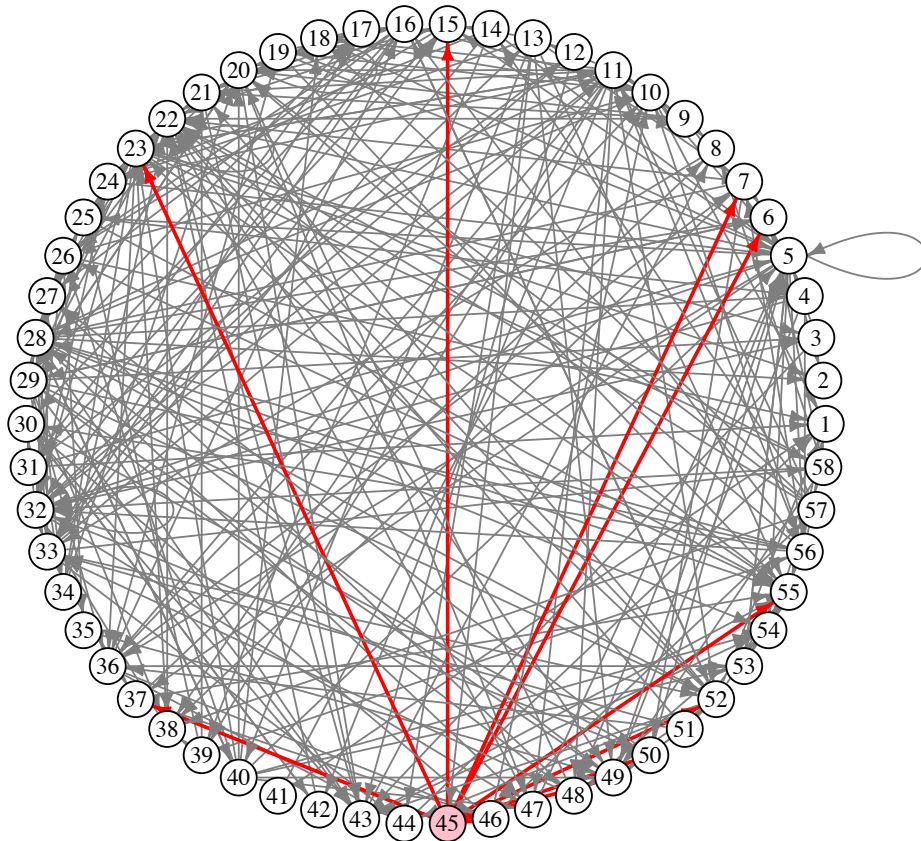


Figure 3.4: Graph of T-cell activation GRN formed by 58 genes. Each gene is represented as a node, and arrows represent the direction of influence. FYB (gene 45) is marked in pink, with its connections highlighted in red.

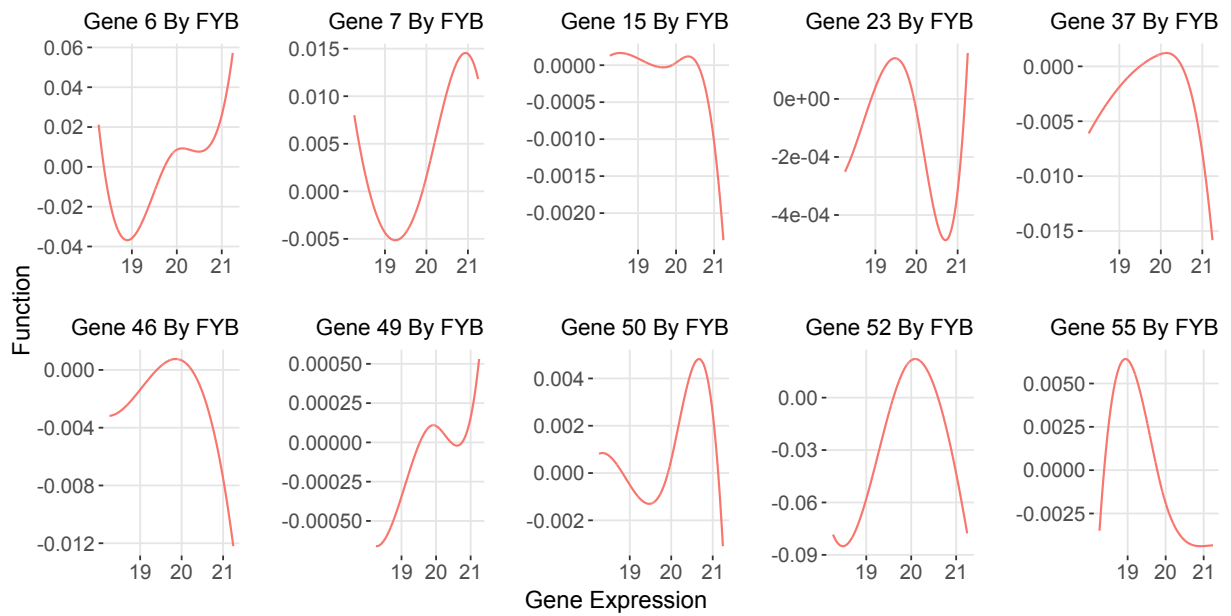


Figure 3.5: The estimated regulation functions of genes influenced by FYB (gene 45). In each panel, the x-axis represents the estimated expression level of FYB, and the y-axis measures the estimated regulation effect, which varies with the estimated expression level of FYB.

Chapter 4

A Bayesian Collocation Integral Method for Model Validation of Ordinary Differential Equations

4.1 Introduction

There may be several competing function forms for $\mathbf{f}$ in (1.3) that exhibit similar dynamics. For example, to study the interaction between predator and prey populations, one might use the basic Lotka–Volterra model (Lotka, 1932), which assumes exponential growth of the prey, while another might assume logistic growth and employ a logistic prey model. ODEs can fail in capturing the full variability in observed real data, leading to discrepancies between model outputs and real processes (Kim, 2016). Measurement error from noisy observations can cause discrepancies because ODE trajectories are inherently deterministic. However, discrepancies can also arise from model inadequacy (Kennedy and O’Hagan, 2001), which may be defined as the difference between the true mean value of the real process and the model output $\mathbf{x}$ at the true parameter values $\boldsymbol{\theta}$. In such cases, validating the ODE model is essential for drawing reliable inferences and making robust predictions.

Considerable effort has been devoted to avoiding ODE model inadequacies and identifying the appropriate models. Kennedy and O’Hagan (2001) formally introduced a model inadequacy function $\mathbf{d}(t_j)$ to the original measurement model (1.2) to systematically measure the model inadequacy,

$$\mathbf{y}_{js} = \mathbf{x}(t_j) + \boldsymbol{\epsilon}_{js} + \mathbf{d}(t_j), \quad j = 1, \dots, J, s = 1, \dots, S, \quad (4.1)$$

where $\mathbf{y}_{js} = (\mathbf{y}_{1js}, \dots, \mathbf{y}_{Ijs})^\top$ is a vector of noisy observation for the s th replication at time point j , $\mathbf{d}(t_j) = (d_1(t_j), \dots, d_I(t_j))^\top$ denotes an unspecified function representing the potential model inadequacy of $\mathbf{y}_j$ at time point j . If this function and its uncertainty can be estimated from data, the validity of the model can be evaluated. Bayarri et al. (2007) pointed out that since the model parameters and inadequacy function are highly confounded, they should be simultaneously estimated. From a frequentist perspective, one approach is to model the inadequacy function as a nonlinear smooth function of time when the ODE parameters are fixed. For independently measured time-course data, Kim (2016) proposed a likelihood-based validation method to estimate the inadequacy function by fitting smoothing splines. Following this idea, Kim and Kim (2019) addressed complex structures in time-course data, including correlations between objects, time dependence, and heteroscedasticity, and proposed a validation method based on likelihood inferences. Nevertheless, quantifying the uncertainty of inadequacy function and model parameters in these likelihood-based approaches remains challenging.

An alternative approach to ODE model validation is the Bayesian framework, which can better quantify the uncertainty of estimates (Wang et al., 2022). To validate simple dynamic networks, Rempala et al. (2007) introduced a Bayesian method that imposes a multivariate Gaussian prior on the inadequacy function, where the covariance structure considers the possible dependencies among the inadequacy functions between and within the trajectories. Kim et al. (2013) extended their methods to discriminate ODE models in the context of gene regulatory networks. However, their approach prespecified the covariance structure of the measurement error distribution, which lacks flexibility and may not always be possible in practice.

In this paper, we aim to improve existing validation methods for general nonlinear ODEs using our Bayesian collocation integral method (Xu et al., 2024). Specifically, we incorporate model inadequacy as a function into the measurement model to systematically capture potential discrepancies between ODE models and observed data. To address the potential issue of confounding between the model parameters and inadequacy function, we simultaneously estimate them by assuming the inadequacy function is a smooth function that can be approximated using basis function expansions, such as cubic B-splines. The ODE solution $\mathbf{x}(t)$ is also approximated by expansions of basis functions, which allows for using integral-based methods on this approximation. We introduce a smoothing parameter to balance the data fit and ODE fidelity by combining the integrated ODE formulation with the likelihood of the observed data. Inspired by Xu et al. (2024), we develop an algorithm for automatic selection of the smoothing parameter and estimation of model inadequacy function. Our approach accommodates generic priors on parameters and supports flexible choices of measurement error distributions. Using a Markov chain Monte

Carlo (MCMC) sampler, we draw samples from the posterior distribution. Our method enables simultaneous estimation of model inadequacy function and parameters for general nonlinear ODEs, while also supporting model validation and uncertainty quantification. We demonstrate the performance of our approach through simulation studies and provide a real data example.

The remainder of the paper is organized as follows. Section 4.2 presents the formulation of our Bayesian method, including a detailed description of the algorithm and computational techniques. Our proposed method is illustrated using simulated data in Section 4.3 and real data in Section 4.4. Finally, Section 4.5 concludes the paper.

4.2 Methodology

4.2.1 Bayesian Structure

Based on the measurement model (4.1) and the assumption $\epsilon_{ijs} \sim N(0, \sigma_i^2)$, the ideal likelihood can be expressed as

$$p(\mathbf{y}|\mathbf{x}(t), \mathbf{d}, \boldsymbol{\sigma}) \propto \left(\prod_{i=1}^I \prod_{j=1}^J \prod_{s=1}^S \sigma_i^2 \right)^{-1/2} \exp \left\{ - \sum_{i=1}^I \left[\sum_{j=1}^J \sum_{s=1}^S \frac{(y_{ijs} - x_i(t_j) - d_i(t_j))^2}{2\sigma_i^2} \right] \right\}. \quad (4.2)$$

As in (2.1) of Section 2.2, we approximate $x_i(t)$ using an expansion of L cubic B-spline basis functions $\Phi(t) = (\phi_1(t), \phi_2(t), \dots, \phi_L(t))^\top$, i.e. $\hat{x}_i(t) = \Phi(t)^\top \mathbf{c}_i$. For model validation, Bayarri et al. (2007) stated that model parameters $\boldsymbol{\theta}$ and inadequacy function $\mathbf{d}_i(t)$ are highly confounded and they should be estimated simultaneously. One solution is to model $\mathbf{d}_i(t)$ as a nonlinear smooth function of time (Rempala et al., 2007; Kim, 2016; Kim and Kim, 2019). In our Bayesian collocation framework, we choose to approximate $\mathbf{d}_i(t)$ using an expansion of L cubic B-spline basis functions $\Phi(t)$ as for $\mathbf{x}_i(t)$, i.e.,

$$\hat{\mathbf{d}}_i(t) = \Phi(t)^\top \mathbf{e}_i, \quad (4.3)$$

where the column vector $\mathbf{e}_i = (e_{i1}, \dots, e_{iL})$ denotes the basis coefficients for component $i = 1, \dots, I$. For simplicity, we use the same number of basis functions and knots for $\mathbf{d}_i(t)$ and $\mathbf{x}_i(t)$. The likelihood function (4.2) can therefore be substituted by the spline-

approximated version

$$p(\mathbf{y}|\mathbf{c}, \mathbf{e}, \boldsymbol{\sigma}) \propto \left(\prod_{i=1}^I \prod_{j=1}^J \prod_{s=1}^S \sigma_i^2 \right)^{-1/2} \exp \left\{ - \sum_{i=1}^I \left[\sum_{j=1}^J \sum_{s=1}^S \frac{(y_{ijs} - \Phi(t_{ij})^\top \mathbf{c}_i - \Phi(t_{ij})^\top \mathbf{e}_i)^2}{2\sigma_i^2} \right] \right\}. \quad (4.4)$$

Following Section 2.2, we assign prior (2.4) to the basis coefficient $\mathbf{c}$ to take advantage of the information from the ODE model (1.3),

$$\begin{aligned} \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \lambda) &\propto \exp \left\{ -\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\hat{x}_i(t) - \int_0^t f_i(\hat{\mathbf{x}}(s)|\boldsymbol{\theta}) ds - \hat{x}_i(0) \right]^2 dt \right\} \\ &= \exp \left\{ -\frac{\lambda}{2} \sum_{i=1}^I \int_{t_1}^{t_J} \left[\Phi(t)^\top \mathbf{c}_i - \int_0^t f_i(\mathbf{c}^\top \Phi(s)|\boldsymbol{\theta}) ds - \Phi(0)^\top \mathbf{c}_i \right]^2 dt \right\}. \end{aligned} \quad (2.4)$$

where $\lambda > 0$ denotes the tuning parameter that controls trade-off between the fidelity to ODE system (1.3) and fit to the measurement model (4.1). To estimate the inadequacy function $\mathbf{d}_i(t)$, we assign a Gaussian prior to each element of the vector $\mathbf{e}_i$,

$$e_{il} \sim N(0, \tau_i^2), \quad l = 1, \dots, L, \quad (4.5)$$

where $\tau_i > 0$ decides the rate of e_{il} approaching zero, which will be discussed in Section 4.2.2. For the remaining parameters, we may assign generic non-informative prior as in 2.2, that is, $\tilde{\pi}_0(\boldsymbol{\theta}) \propto 1$ and $\tilde{\pi}_0(\sigma_i) \propto 1/\sigma_i$ independently.

$$\begin{aligned} \pi(\boldsymbol{\theta}, \mathbf{c}, \mathbf{e}, \boldsymbol{\sigma}|\mathbf{y}, \lambda, \boldsymbol{\tau}) &\propto \pi(\mathbf{y}|\boldsymbol{\theta}, \mathbf{c}, \mathbf{e}, \boldsymbol{\sigma}, \lambda, \boldsymbol{\tau}) \pi(\mathbf{c}|\boldsymbol{\theta}, \boldsymbol{\sigma}, \lambda) \tilde{\pi}_0(\mathbf{e}|\boldsymbol{\tau}) \tilde{\pi}_0(\boldsymbol{\theta}) \tilde{\pi}_0(\boldsymbol{\sigma}) \\ &= p(\mathbf{y}|\mathbf{c}, \mathbf{e}, \boldsymbol{\sigma}) \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \lambda) \tilde{\pi}_0(\mathbf{e}|\boldsymbol{\tau}) \tilde{\pi}_0(\boldsymbol{\theta}) \tilde{\pi}_0(\boldsymbol{\sigma}), \end{aligned} \quad (4.6)$$

4.2.2 Inadequacy Estimation and Model Validation

We now consider setting an appropriate value for $\boldsymbol{\tau}$ to facilitate estimation of the inadequacy function $\mathbf{d}_i(t)$. Kim and Kim (2019) considered an algorithm to iteratively update the parameter $\boldsymbol{\theta}$ and $d_i(t_j)$, respectively. They first estimate $\boldsymbol{\theta}$; then, the inadequacy function at time t_j is computed as $d_i(t_j) = \sum_{s=1}^S y_{ijs}/S - \hat{x}_i(t_j)$ using the estimated $\boldsymbol{\theta}$. We adopt their idea to obtain a preliminary estimate of $\mathbf{d}$ to use for setting the value for $\boldsymbol{\tau}$. We propose a modified smoothing parameter algorithm (Algorithm 2) of Algorithm 1

in Section 2.2.3, which allows joint estimation of $\mathbf{d}$ and $\boldsymbol{\theta}$ via an MCMC sampler based on our posterior distribution (4.6). Specifically, We first obtain the model parameters from the posterior distribution without inadequacy function $\pi(\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma} | \mathbf{y}, \lambda)$, and calculate $d_i^*(t_j) = \sum_{s=1}^S y_{ijs}/S - \hat{x}_i^*(t_j)$ using the model parameters to find the corresponding $\hat{\mathbf{e}}_i^*$ by fitting a smoothing spline. Then $\hat{\boldsymbol{\sigma}}^*$ can be obtained by maximizing the marginal posterior distribution $\pi(\boldsymbol{\sigma} | \hat{\mathbf{e}}_i^*, \boldsymbol{\theta}^*, \hat{\mathbf{c}}^*, \lambda)$. Using $\hat{\boldsymbol{\sigma}}^*$, $\hat{\boldsymbol{\tau}}$ can be obtained by maximizing the marginal posterior distribution $\pi(\boldsymbol{\tau} | \hat{\mathbf{e}}_i^*, \boldsymbol{\theta}^*, \hat{\mathbf{c}}^*, \hat{\boldsymbol{\sigma}}^*, \lambda)$. With this value $\hat{\boldsymbol{\tau}}$, we can infer all parameters simultaneously via the joint posterior distribution (4.6) $\pi(\boldsymbol{\theta}, \mathbf{c}, \mathbf{e}, \boldsymbol{\sigma} | \mathbf{y}, \lambda, \hat{\boldsymbol{\tau}})$.

Similarly to Algorithm 1, Algorithm 2 starts with a small λ , repeats the aforementioned steps, and increases λ by a factor of 10 iteratively. In this algorithm, the stopping rule is the same as the one in Algorithm 1, which considers the inference of parameter estimates $\boldsymbol{\theta}$, the discrepancy between the observations and the estimated ODE solution, and the discrepancy between the B-spline approximation $\hat{\mathbf{x}}(t)$ and the estimated ODE solution. Notice that the inadequacy function is not considered in the stopping rule; it is only used for model validation.

Regarding the model inference and validation, the final MCMC samples of $\boldsymbol{\theta}, \mathbf{c}, \mathbf{e}$ and $\boldsymbol{\sigma}$ are those associated with the output value $\hat{\lambda}$ chosen by Algorithm 2. To validate the model, we consider the central 95% pointwise credible intervals of the individual inadequacy function $\hat{\mathbf{d}}_i(t) = \Phi(t)^\top \hat{\mathbf{e}}_i$. As long as the credible intervals for all $i = 1, \dots, I$ include zero over $[t_1, t_J]$, we conclude that this ODE model is uniformly valid for observed time-course data (Kim and Kim, 2019). To ensure consistency in model validation and selection, we use a single model to select λ and apply the same value across all models.

Following 2.2.3, we recommend setting $\alpha = 0.1$; in other words, using $1 - \alpha = 0.9$ as the threshold for the ratio of overlaps between the credible intervals of the parameters for different λ . When observations are dense, we start with $\lambda^{(0)} = 10^2$ and set $\lambda^* = 10^3$; when observations are relatively sparse, we start with $\lambda^{(0)} = 10$ and set $\lambda^* = 10^2$.

4.2.3 Model Selection

When two or more models are considered valid, it may be of interest to select the most appropriate one. We consider comparing the Bayes factor (BF, Kass and Raftery, 1995) criterion based on the posterior distribution of model without inadequacy function (Kim et al., 2013), i.e.,

$$\begin{aligned} \pi(\dot{\boldsymbol{\theta}}, \dot{\mathbf{c}}, \dot{\boldsymbol{\sigma}} | \mathbf{y}, \dot{\lambda}) &\propto \pi(\mathbf{y} | \dot{\boldsymbol{\theta}}, \dot{\mathbf{c}}, \dot{\boldsymbol{\sigma}}, \dot{\lambda}) \pi(\dot{\mathbf{c}} | \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\sigma}}, \dot{\lambda}) \tilde{\pi}_0(\dot{\boldsymbol{\theta}}) \tilde{\pi}_0(\dot{\boldsymbol{\sigma}}) \\ &= p(\mathbf{y} | \dot{\mathbf{c}}, \dot{\boldsymbol{\sigma}}) \tilde{\pi}_0(\dot{\mathbf{c}} | \dot{\boldsymbol{\theta}}, \dot{\lambda}) \tilde{\pi}_0(\dot{\boldsymbol{\theta}}) \tilde{\pi}_0(\dot{\boldsymbol{\sigma}}), \end{aligned}$$

Algorithm 2 Smoothing parameter algorithm

- 1: **Inputs:** a small initial $\lambda^{(0)}$; a moderately large threshold λ^* after which the stopping criterion starts to be checked; the level α for the overlap ratio of the credible interval; initial estimates $\hat{\boldsymbol{\theta}}(\lambda^{(0)})$, $\hat{\mathbf{c}}(\lambda^{(0)})$ and $\hat{\boldsymbol{\sigma}}(\lambda^{(0)})$ corresponding to $\lambda^{(0)}$
 - 2: **Output:** the final choice $\hat{\lambda}$
 - 3: Set $p = 1$, $\lambda^{(1)} = \lambda^{(0)} \times 10$
 - 4: **while** $\lambda^{(p)} \leq 10^6$ **do**
 - 5: Sample $\boldsymbol{\theta}$, $\mathbf{c}$ and $\boldsymbol{\sigma}$ from the posterior distribution $\pi(\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma} | \mathbf{y}, \lambda^{(p)})$ given by (2.5), using the values $\hat{\boldsymbol{\theta}}(\lambda^{(p-1)})$, $\hat{\boldsymbol{\sigma}}(\lambda^{(p-1)})$ and $\hat{\mathbf{c}}(\lambda^{(0)})$ to initialize the MCMC sampler
 - 6: Using the MCMC samples, compute the posterior means $\hat{\boldsymbol{\theta}}^*(\lambda^{(p)})$ and $\hat{\mathbf{c}}^*(\lambda^{(p)})$ for $\boldsymbol{\theta}$ and $\mathbf{c}$
 - 7: Calculate the inadequacy function $d_i^*(t) = \sum_{s=1}^S y_{is}(t)/S - \boldsymbol{\Phi}(t)^\top \hat{\mathbf{c}}^*(\lambda^{(p)})$ and fit smoothing splines to find the corresponding $\hat{\mathbf{e}}_i^*(\lambda^{(p)})$
 - 8: Using the MCMC samples, obtain $\hat{\boldsymbol{\sigma}}^*(\lambda^{(p)})$ by maximizing the marginal posterior distribution $\pi(\boldsymbol{\sigma} | \hat{\mathbf{e}}_i^*(\lambda^{(p)}), \hat{\boldsymbol{\theta}}^*(\lambda^{(p)}), \hat{\mathbf{c}}^*(\lambda^{(p)}), \lambda^{(p)}, \mathbf{y})$
 - 9: Using the MCMC samples, obtain $\hat{\boldsymbol{\tau}}(\lambda^{(p)})$ by maximizing the marginal posterior distribution $\pi(\boldsymbol{\tau} | \hat{\mathbf{e}}_i^*(\lambda^{(p)}), \hat{\boldsymbol{\theta}}^*(\lambda^{(p)}), \hat{\mathbf{c}}^*(\lambda^{(p)}), \hat{\boldsymbol{\sigma}}^*(\lambda^{(p)}), \lambda^{(p)}, \mathbf{y})$
 - 10: Sample $\boldsymbol{\theta}$, $\mathbf{c}$, $\mathbf{e}$ and $\boldsymbol{\sigma}$ from the posterior distribution $\pi(\boldsymbol{\theta}, \mathbf{c}, \mathbf{e}, \boldsymbol{\sigma} | \mathbf{y}, \lambda^{(p)}, \boldsymbol{\tau}(\lambda^{(p)}))$ given by (4.6), using the values $\hat{\boldsymbol{\theta}}(\lambda^{(p-1)})$, $\hat{\boldsymbol{\sigma}}(\lambda^{(p-1)})$, $\hat{\mathbf{e}}_i^*(\lambda^{(p)})$ and $\hat{\mathbf{c}}(\lambda^{(0)})$ to initialize the MCMC sampler
 - 11: Compute $\hat{\boldsymbol{\theta}}(\lambda^{(p)})$, $\hat{\mathbf{c}}(\lambda^{(p)})$, $\hat{\mathbf{e}}(\lambda^{(p)})$ and $\hat{\boldsymbol{\sigma}}(\lambda^{(p)})$ for $\boldsymbol{\theta}$, $\mathbf{c}$, $\mathbf{e}$ and $\boldsymbol{\sigma}$, and the central 95% credible interval for each parameter in $\boldsymbol{\theta}$
 - 12: Calculate $Err(\lambda^{(p)})$ in (2.8) which measures the discrepancies between the observations and the estimated ODE solution, and between the B-spline approximation and the estimated ODE solution.
 - 13: **if** $\lambda^{(p)} \geq \lambda^*$ **then**
 - 14: **if** $Err(\lambda^{(p)}) \leq Err(\lambda^{(p-1)})$ **then**
 - 15: For each parameter in $\boldsymbol{\theta}$, calculate the overlap of its credible intervals corresponding to $\lambda^{(p)}$ and $\lambda^{(p-1)}$
 - 16: **if** the ratio of the overlap to the current credible interval for each parameter in $\boldsymbol{\theta}$ is larger than $1 - \alpha$ **then**
 - 17: **return** $\hat{\lambda} = \lambda^{(p)}$
 - 18: **end if**
 - 19: **else if** $Err(\lambda^{(p)}) > Err(\lambda^{(p-1)})$ **then**
 - 20: **return** $\hat{\lambda} = \lambda^{(p-1)}$
 - 21: **else if** $\lambda^{(p)} = 10^6$ **then**
 - 22: **return** $\hat{\lambda} = \lambda^{(p)}$
 - 23: **end if**
 - 24: **end if**
 - 25: $\lambda^{(p+1)} = \lambda^{(p)} \times 10$
 - 26: $p \leftarrow p + 1$
 - 27: **end while**
-

where $\dot{\lambda}$, $\dot{\theta}$, $\dot{c}$, and $\dot{\sigma}$ denote the MCMC samples obtained using the final λ selected by Algorithm 2. In the case where the inadequacy function $d = \mathbf{0}$, then sampling from this posterior distribution is equivalent to sampling from (4.6). The R package *bridgesampling* (Gronau et al., 2020) is applied to calculate the BFs of one model over another.

4.2.4 Practical Implementation

Similarly to Section 2.2.4 and Section 3.2.4, we implement the posterior distribution (4.6) in C++ and pass it to *tmbstan* (Monnahan and Kristensen, 2018) with NUTS (Homan and Gelman, 2014) as the sampling algorithm. As this posterior distribution contains two integrals that are not usually expressed in closed form, the Gaussian quadrature rule (Stroud and Secrest, 1966) is applied to approximate them, and more details of the approximation including the choice for the number of quadrature points can be found in Section 2.2.2. In all of the subsequent examples, one chain of 1000 iterations with a 500-iteration warmup is run for our algorithm. More details on the implementation can be found in Section A.2 of the appendix.

To set the initial values for θ , c and σ , we use the same method as in Section 2.2.4. First, the R package *CollocInfer* (Hooker et al., 2016) is applied to smooth the data with the roughness penalty 0.1 and the coefficients are extracted as $\hat{c}(\lambda^{(0)})$. Then, we set $\hat{\sigma}_i(\lambda^{(0)}) = 0.1$. With other parameters fixed, we plug $\hat{c}(\lambda^{(0)})$ into (2.4). Using the draws from one 1000-iteration chain with a 500-iteration warmup, the posterior mean of θ is treated as the initial estimate $\hat{\theta}(\lambda^{(0)})$.

4.3 Simulation

4.3.1 Benchmark System and Setup

In this simulation study, we evaluate our method using the example from Kim (2016) with two models which can produce similar trajectories:

$$\text{Model 1: } \mathbf{f}_1(\mathbf{x}(t), \theta, t) = \begin{bmatrix} \theta_1 - \theta_2 x_1 \\ \theta_3 x_1 - \theta_4 x_2 \end{bmatrix}, \quad (4.7)$$

$$\text{Model 2: } \mathbf{f}_2(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} \frac{\theta_1 x_2}{1 + x_2} - \theta_2 x_2 \\ \theta_3 x_1 - \frac{\theta_4 x_2}{1 + x_2} \end{bmatrix}, \quad (4.8)$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)$ and $\mathbf{x}(t) = (x_1(t), x_2(t))^\top$. Following the set up in [Kim \(2016\)](#), we treat Model 1 as the true model and Model 2 as an inadequate model, i.e., $\mathbf{d} \neq \mathbf{0}$ in Model 2 ¹⁴. The true values of the parameters are set as $\boldsymbol{\theta} = (0.5, 0.5, 1, 1)$ and the initial values are set as $x_1(0) = 0$ and $x_2(0) = 1$. We generate 11 equally spaced noisy observations and 3 replications ($S = 3$) of x_1 and x_2 over the time interval $[0, 10]$, using two noise settings for both components: $\sigma_i = 0.02$ (low noise) and $\sigma_i = 0.05$ (high noise). For both noise levels, we place 11 equally-spaced knots with 13 basis functions within the time interval. To demonstrate inference and validation results without specific prior knowledge of the parameters, we assign uniform priors over the positive real numbers to each parameter in $\boldsymbol{\theta}$. When implementing the smoothing parameter algorithm 2, we use Model 2 to select λ and apply it in Model 1. Since the observations are relatively dense in the time interval, we start from $\lambda^{(0)} = 10^2$ with $\lambda^* = 10^3$. Following the guideline in 2.2.2, we apply $M_i = 25$ and $K_{m_i} = 4$ quadrature points for the outer and inner integral, respectively.

Since we do not have access to the code for the methodology in [Kim \(2016\)](#), we do not include it in our comparison. Our settings also differ, as they consider ODE models for time-course data with heteroscedasticity and their evaluation is only based on a single simulation run. Furthermore, their method can be computationally intensive when applied to large systems. The confidence bands for the inadequacy function in [Kim \(2016\)](#) need to be estimated using a Monte Carlo method and requires substantive numerical optimizations, whereas our Bayesian framework directly provides uncertainty quantification for both model inadequacy function and parameters.

To evaluate the variability of parameter estimates and model validation, 100 independent simulation trials are run for both noise settings. In all simulation studies, we numerically solve differential equations using the R package *deSolve* ([Soetaert et al., 2010](#)).

¹⁴For demonstration, we use the same setup as in [Kim \(2016\)](#). If Model 2 is treated as the true model and Model 1 as an inadequate one, we find that our method still works empirically.

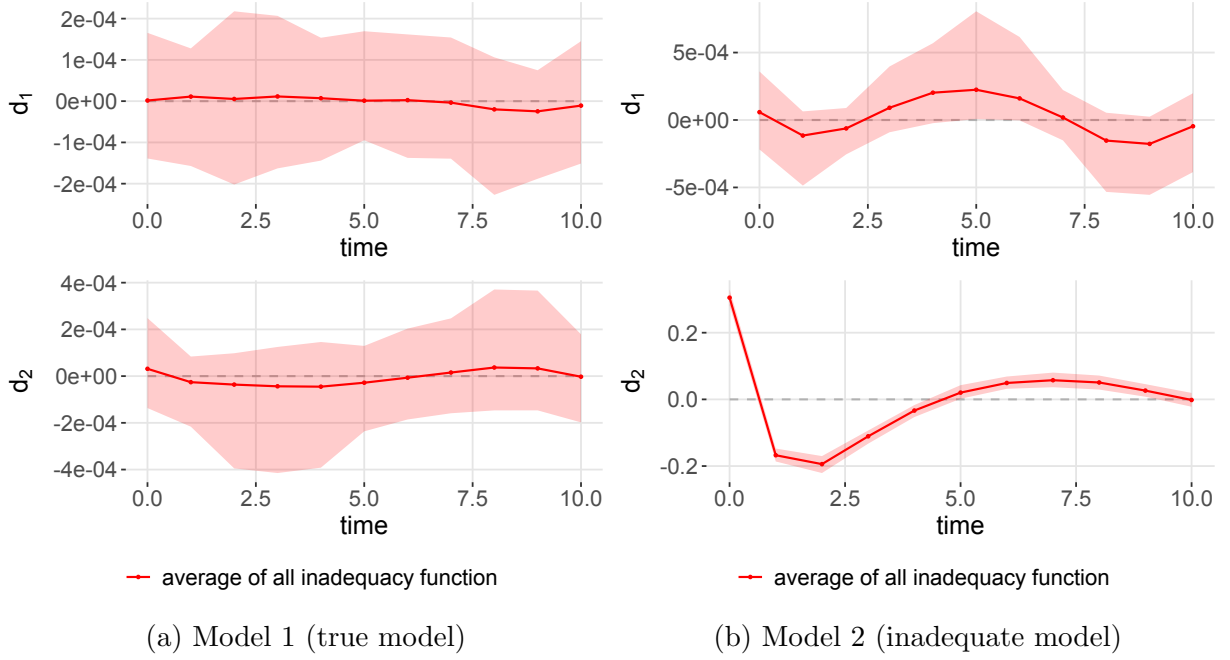


Figure 4.1: Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with low-noise settings. The black dashed line represents zero line. The red solid line represents the average estimated inadequacy function, and the red shaded area represents the pointwise central 95% intervals of all estimated inadequacy functions.

4.3.2 Performance Assessment

As in Section 2.3.2, we consider two metrics to evaluate the performance of the parameter estimates and their associated trajectories for the true model. For parameter estimation, we compute the root mean square error (RMSE) relative to the true parameter values. For system recovery, we evaluate the trajectory RMSE, defined as

$$\text{trajectory RMSE} = \sqrt{\int_{t_1}^{t_J} [\tilde{x}(t) - x(t)]^2 dt},$$

where $\tilde{x}(t)$ and $x(t)$ denote the reconstructed and true trajectories, respectively. The integral inside is approximated by a Riemann sum over a dense grid of time points. Notice that a numerical solver is applied to compute $\tilde{x}(t)$ and $x(t)$ solely for performance evaluation; it is not required for implementing our method. Further details on these two performance

metrics are provided in Section 2.3.2.

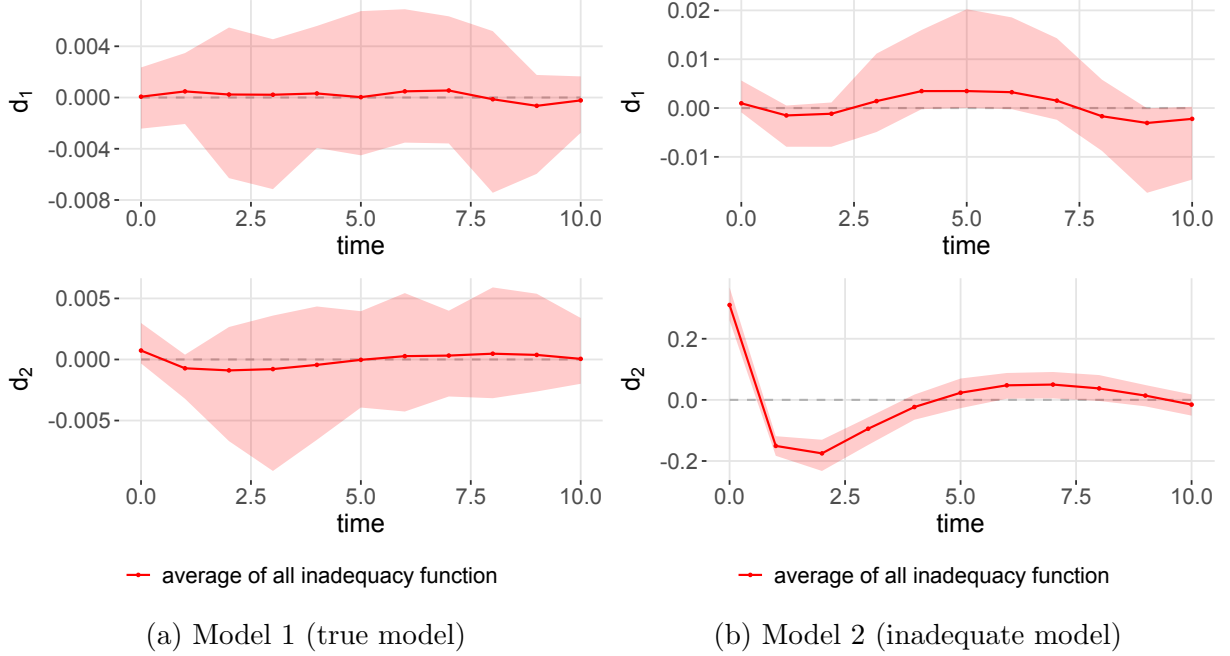


Figure 4.2: Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with high-noise settings. The black dashed line represents zero line. The red solid line represents the average estimated inadequacy function, and the red shaded area represents the pointwise central 95% intervals of all estimated inadequacy functions.

4.3.3 Results

Figures 4.1 and 4.2 show the average estimated inadequacy functions across 100 simulation runs, along with the 95% central pointwise intervals under low and high noise levels, respectively. With a higher noise level, we can expect that the 95% central intervals can be slightly wider. Despite this, the estimated inadequacy functions of the true model remain very close to zero for both the components and the noise settings, with relatively narrow 95% pointwise credible intervals. In contrast, the estimated inadequacy functions of the inadequate model deviate noticeably from zero line, particularly for component x_2 . In all 100 simulation runs across both noise settings, the true model is consistently treated as valid, whereas the inadequate model is consistently treated as invalid. Therefore, we can conclude that our method performs well in model validation for this simulation.

The results of parameter estimation for Model 1 are summarized in Table 4.1, while Table 4.2 presents the median and interquartile range (IQR) of trajectory RMSEs for both models. Although parameter RMSEs are slightly higher under the high-noise setting, our method provides reliable parameter estimates across both noise levels. Moreover, based on trajectory RMSEs under different noise levels, it accurately reconstructs the ODE solution for the true model. On the other hand, the inadequate model has dramatically higher trajectory RMSEs as expected.

Similar to Section 2.3.3, Figure 4.3 and Figure 4.4 display the average estimated ODE solution across 100 simulation runs for low and high noise levels, respectively, along with the corresponding 95% pointwise intervals. Although the intervals are slightly wider under the high-noise setting, our method successfully recovers the system and provides reliable inference for the ODE solution in both cases.

We also evaluate the sensitivity of τ by varying its value when we sample θ , $\mathbf{c}$, $\mathbf{e}$ and σ from the posterior distribution $\pi(\theta, \mathbf{c}, \mathbf{e}, \sigma | \mathbf{y}, \lambda, \tau)$ given by (4.6). The results show that our method is not particularly sensitive to the value set for τ . Further details are provided in Section D.1.1 of the appendix.

Noise Level	θ_1	θ_2	θ_3	θ_4
Low-noise, $\sigma = 0.02$	0.500(0.010)	0.500(0.012)	1.010(0.049)	1.011(0.050)
High-noise, $\sigma = 0.05$	0.499(0.025)	0.498(0.030)	1.003(0.103)	1.002(0.105)

Table 4.1: Mean of the parameter estimates with RMSE in parentheses for Model 1 (4.7) across 100 simulated runs.

Noise Level	Model	x_1	x_2	Total
Low-noise, $\sigma = 0.02$	1 (True)	0.006(0.003)	0.004(0.002)	0.008(0.003)
	2 (Biased)	0.086(0.007)	0.202(0.007)	0.219(0.008)
High-noise, $\sigma = 0.05$	1 (True)	0.013(0.004)	0.011(0.003)	0.019(0.006)
	2 (Biased)	0.094(0.015)	0.214(0.014)	0.232(0.015)

Table 4.2: Median of trajectory RMSEs (with IQR in parentheses) for each component of Model 1 (4.7) and Model 2 (4.8) across 100 simulated runs.

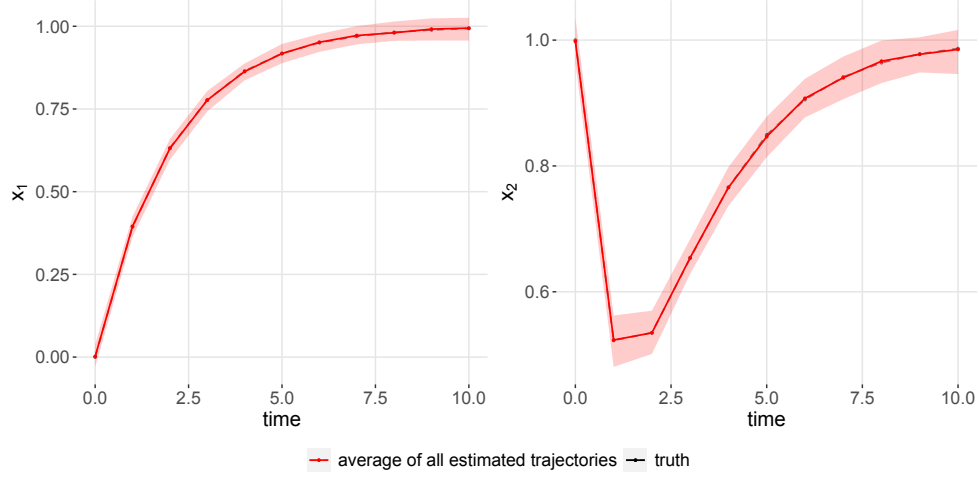


Figure 4.3: Estimated trajectories for x_1 (left panel) and x_2 (right panel) in Model 1 (4.7), over 100 simulation runs with low-noise settings. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

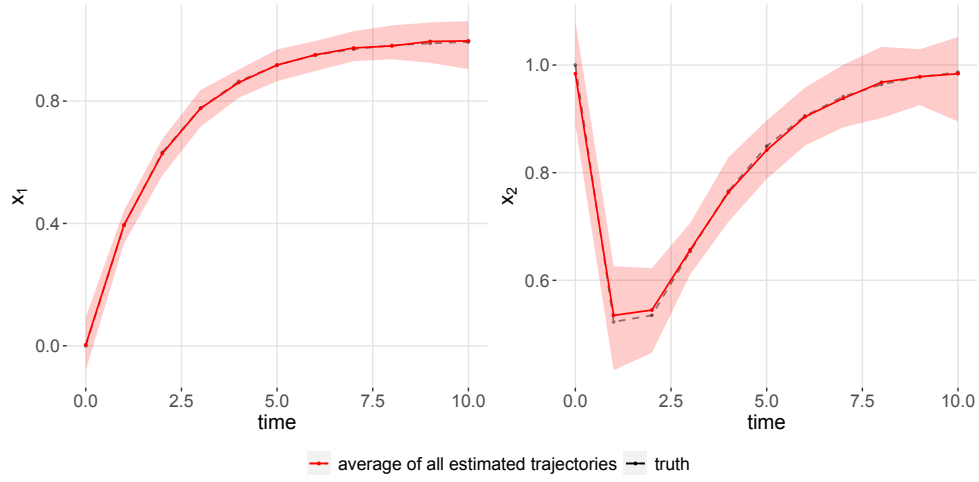


Figure 4.4: Estimated trajectories for x_1 (left panel) and x_2 (right panel) in Model 1 (4.7), over 100 simulation runs with high-noise settings. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

4.4 Real Data Analysis

Gause (1935) designed an experiment to investigate the interaction between *Paramecium aurelia* and *Paramecium caudatum*. The experiment maintained the two species in a constant-volume physiological salt solution, with a fixed amount of bacterial culture added daily. All measurements were initialized with 20 individuals per species for three replicates. The data is provided in Table 3 of the appendix in Gause (1935), where the first sample census was taken when each replicate was two days old, and each surviving replicate was sampled daily thereafter until the nineteenth day after the start of the experiments.

To explore the relationship between two species, Lotka-Volterra (LV, Lotka, 1932) competition model was repeatably used for this experiment (Gause, 1935; Leslie, 1957),

$$\text{Model 1: } \mathbf{f}_1(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} \psi_1 x_1 - \psi_2 x_1 x_2 - \psi_3 x_1^2 \\ \psi_4 x_2 - \psi_5 x_1 x_2 - \psi_6 x_2^2 \end{bmatrix}, \quad (4.9)$$

where the parameter vector is $\boldsymbol{\theta} = (\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6)$. We assume x_1 denotes *Paramecium aurelia* and x_2 is *Paramecium caudatum*. Murdoch et al. (2013) provided a detailed review for some competing LV predator-prey type models and here we consider four candidate models of them,

$$\text{Model 2: } \mathbf{f}_1(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} \psi_2 \psi_3 x_1 x_2 - \psi_4 x_1 \\ \psi_1 x_2 - \psi_2 x_1 x_2 \end{bmatrix}, \quad (4.10)$$

$$\text{Model 3: } \mathbf{f}_2(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} \psi_2 \psi_3 x_1 x_2 - \psi_4 x_1 \\ \psi_1 x_2 \left(1 - \frac{x_2}{\psi_5}\right) - \psi_2 x_1 x_2 \end{bmatrix}, \quad (4.11)$$

$$\text{Model 4: } \mathbf{f}_3(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} \frac{\psi_2 \psi_3 x_1 x_2}{1 + \psi_2 \psi_5 x_1} - \psi_4 x_1 \\ \psi_1 x_2 - \frac{\psi_2 x_1 x_2}{1 + \psi_2 \psi_5 x_1} \end{bmatrix}, \quad (4.12)$$

$$\text{Model 5: } \mathbf{f}_4(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} \psi_2 \psi_3 x_1 x_2 - \psi_4 x_1 - \psi_5 x_1^2 \\ \psi_1 x_2 - \psi_2 x_1 x_2 \end{bmatrix}, \quad (4.13)$$

where the parameter vector is $\boldsymbol{\theta} = (\psi_1, \psi_2, \psi_3, \psi_4, \psi_5)$ for Model 3, 4 and 5 and $\boldsymbol{\theta} =$

$(\psi_1, \psi_2, \psi_3, \psi_4)$ for Model 2. We assume x_1 denotes *Paramecium aurelia* (predator) and x_2 is *Paramecium caudatum* (prey).

In the absence of prior information, all parameters in θ are assigned uniform distributions on $(0, \infty)$. We place 19 knots only at each observed time point $\{0, 2, \dots, 19\}$ with 21 cubic B-spline basis. Given the relative sparsity of observations in this interval, we set $\lambda^{(0)} = 10$ and $\lambda^* = 10^2$ in Algorithm 2, and use Model 4 to select λ . To approximate the outer and inner integrals, we apply $M_i = 40$ and $K_{m_i} = 4$ quadrature points, respectively, following the guideline in Section 2.2.2. All other settings for implementing our method are the same as those used in simulation studies.

Using Algorithm 2, Model 4 selects $\hat{\lambda} = 10$ as the final value. The estimated inadequacy functions with their corresponding 95% pointwise credible intervals are shown in Figures 4.5 and 4.6, where Models 2 and 3 are considered invalid, as their credible intervals do not fully cover the zero line. Although Models 4 and 5 are considered valid, their estimated inadequacy functions deviate more from the zero line than that of Model 1. As depicted in Figure 4.7, Models 1 and 4 seem to give a more reasonable fit for the data than Model 5 based on estimated trajectories. However, it is still difficult to determine the better model between Models 1 and 4. As stated in Section 4.2.3, we calculate Bayes factors (BFs) to select the best model among Models 1, 4, and 5. Based on the BFs comparing Model 1 to Model 4 and Model 1 to Model 5, Model 1 is strongly preferred, with factors exceeding 10,000. Given that the average mean square bias for Models 1, 4, and 5 is 19.789, 89.304, and 646.455, respectively, Model 1 is expected to be superior, as it provides a better fit to the data. This result is consistent with the findings in Gause (1935), as *Paramecium aurelia* and *Paramecium caudatum* are known to be competitors rather than predator and prey. Table 4.3 summarizes the parameter estimation results, further indicating that the inadequacy function estimation is not particularly sensitive to the estimation of the hyperparameter τ .

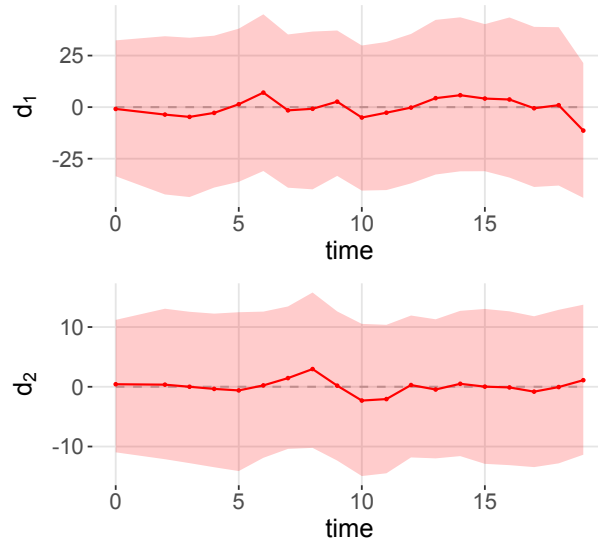
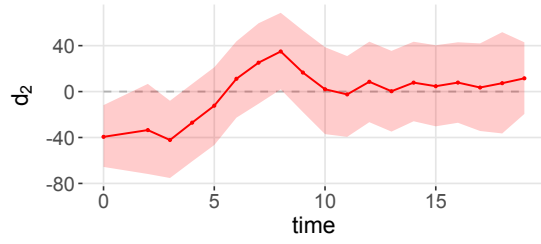
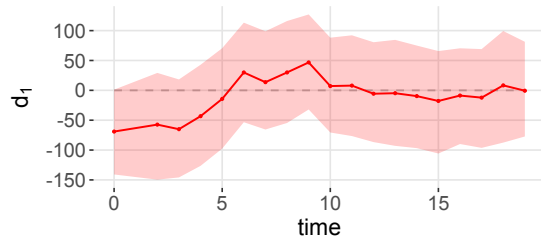
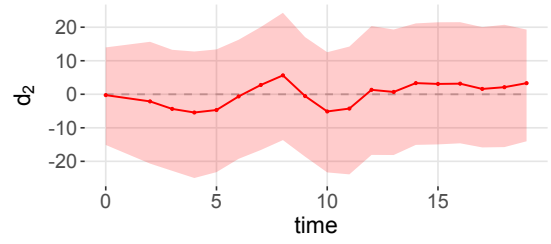
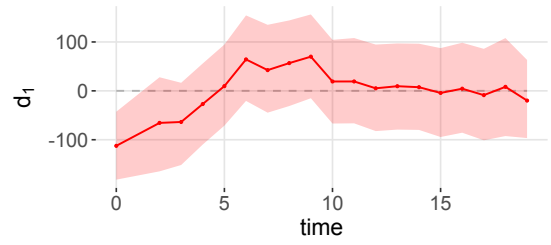


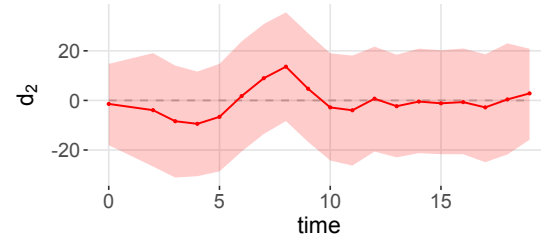
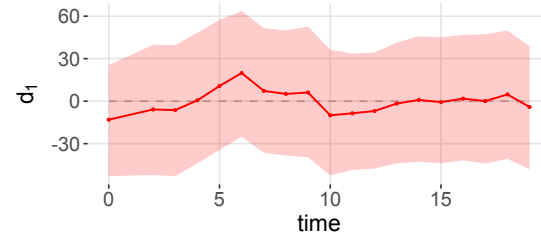
Figure 4.5: Estimated inadequacy functions for the population of *Paramecium aurelia* (x_1) and *Paramecium caudatum* (x_2) in a 19-day period for LV competition model. The black dashed line represent the zero line and the red solid line is the estimated inadequacy function. The red shaded area is a central 95% pointwise credible interval for the estimated inadequacy function.



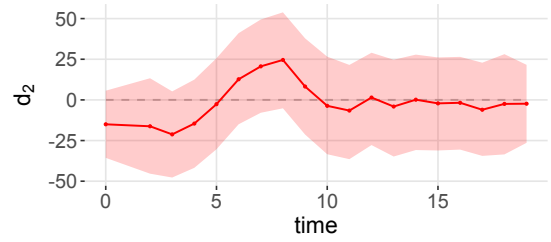
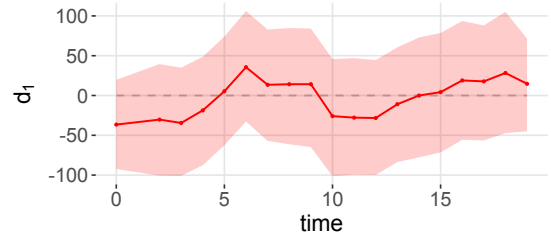
(a) Model 2



(b) Model 3



(c) Model 4



(d) Model 5

Figure 4.6: Estimated inadequacy functions for the population of *Paramecium aurelia* (x_1) and *Paramecium caudatum* (x_2) in a 19-day period for four LV predator-prey type models. The black dashed line represent the zero line and the red solid line is the estimated inadequacy function. The red shaded area is a central 95% pointwise credible interval for the estimated inadequacy function.

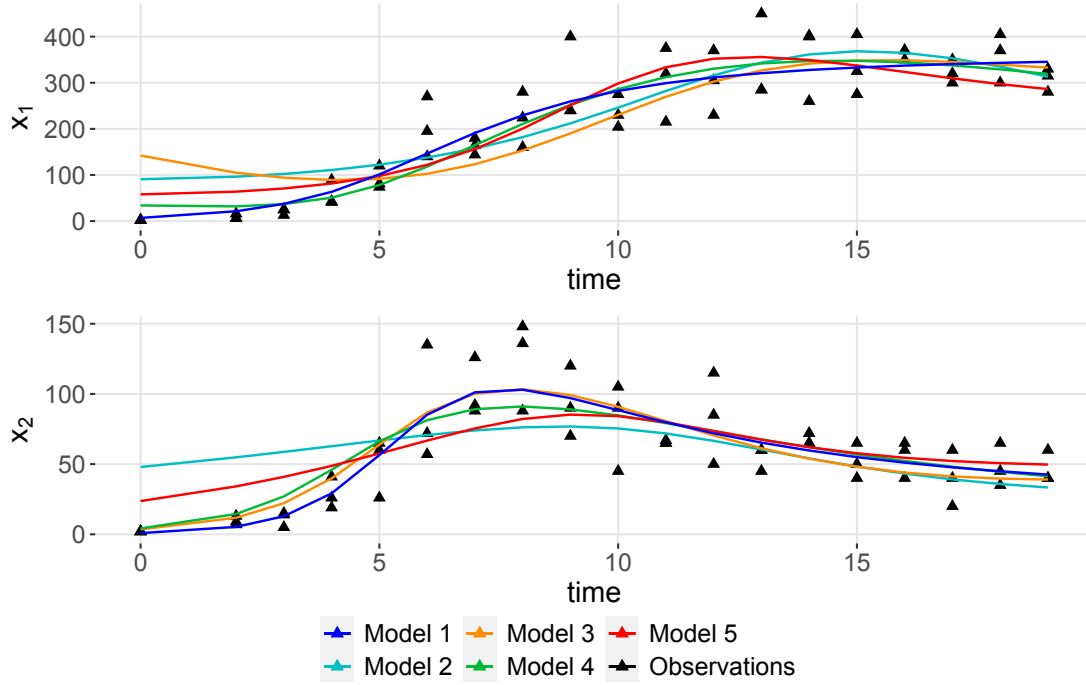


Figure 4.7: Estimated trajectories for the population of *Paramecium aurelia* (x_1) and *Paramecium caudatum* (x_2) in a 19-day period for five LV type models. The black triangles represent the real data. The blue, light blue, orange, green and red solid lines are the estimated trajectories for Model 1, 2, 3, 4 and 5, respectively.

	Model 1	Model 2	Model 3	Model 4	Model 5
θ_1	0.674	0.148	1.056	2.012	0.255
θ_2	0.001	0.001	0.002	0.113	0.001
θ_3	0.002	27.926	1.816	0.545	7.983
θ_4	1.124	0.347	0.179	0.153	0.059
θ_5	0.003	-	161.706	0.528	0.001
θ_6	0.005	-	-	-	-
τ_1	410.908	3170.302	4096.386	612.218	1285.491
τ_2	46.225	678.038	110.456	167.268	1279.710

Table 4.3: Parameter estimates results using five LV type models to fit the population of *Paramecium aurelia* (x_1) and *Paramecium caudatum* (x_2) in a 19-day period.

4.5 Discussion

In this paper, we propose a Bayesian collocation integral method that incorporates model inadequacy as a function into the measurement model for ODE model validation and parameter estimation simultaneously. Compared to existing approaches, our method is applicable to general nonlinear ODEs, and the Bayesian framework enables more effective uncertainty quantification for both model parameters and inadequacy functions. We demonstrate the performance of the proposed method through simulation studies focused on model validation.

We examine the stability of our method through simulation studies under varying noise levels. In practice, there might be two or more models that are considered valid, indicating the need for further investigation into model selection strategies. Therefore, we adopt the Bayes factor criterion to select the most appropriate model and yield satisfactory results.

Chapter 5

Discussion and Future Work

In this thesis, we propose Bayesian integral-based methods for parameter estimation in ODE systems with both known and unknown functional forms, as well as for model validation in systems with known functions. Specifically, we consider incorporating the integral form of ODEs into our proposed model, which avoids derivative estimation. In contrast to the frequentist methods, the Bayesian framework can better quantify the parameter uncertainty. Simulation studies demonstrate that the proposed methods perform favourably, compared to existing ones, in terms of estimation accuracy and model validation.

In Chapter 2, we first consider the ODEs with known functional forms. The proposed method can be applied to general nonlinear ODEs and can automatically select the smoothing parameter. Compared to the Bayesian derivative-based methods published prior to our work [Xu et al. \(2024\)](#), our integral-based method provides superior performance in terms of inference accuracy. Simulation studies examine the stability of our method under different sampling frequencies. Through the comparison of parameter estimation and trajectory recovery, we find a phase transition between the Bayesian collocation derivative-based method and our proposed integral-based method. Specifically, when the observations are sufficiently dense or extremely sparse, only a minor difference between the two methods is displayed, but the integral-based method remains superior. For moderately sparse observations, the advantage of the integral-based method will become particularly evident, especially in recovering the trajectories. Nevertheless, our method can be limited to systems without entirely unobserved components. We leave an extension of our work to systems with completely unobserved components for future research.

In Chapter 3, we extend our method to the ODEs with unknown functional forms to identify the system structure. In particular, we consider the generalized sparse additive

ODE systems. Different from the framework in Chapter 2, the spike-and-slab group lasso penalty is introduced to our model to identify the sparse ODE structure and estimate the additive component simultaneously. Through the simulation studies, we examine the stability of our method under different noise levels and sampling frequencies in terms of trajectory recovery and additive component estimation. Our method can provide reliable inferences for system recovery regarding the pointwise coverage probabilities of trajectories. Moreover, we demonstrate the applicability of our method by constructing a nonlinear dynamic GRN from noisy T-cell activation gene expression data. We also briefly discuss the potential of our proposed method for higher-order ODEs using Green’s matching (Tan et al., 2024), leaving the practical implementation for future research.

In Chapter 4, we consider the case of several competing functional forms for ODEs that exhibit similar dynamics. To address potential model inadequacies and identify the appropriate ODE model, we incorporate model inadequacy as a function into the measurement model. The proposed method enables simultaneous model validation and parameter estimation, and we examine its stability under varying noise levels through simulation studies. Moreover, when two or more models are treated as valid, we adopt the Bayes factor criterion to select the most appropriate one.

In this thesis, we used at most 1,000 MCMC iterations to perform simulation studies and real data analyses. The small number of iterations is due to computational challenges, and a more efficient MCMC implementation is left for future work. In future research, we may also exploit the sequential nature of the problem, for example, by applying a Laplace approximation to the integral contained in the priors.

References

- Abegaz, F. and E. Wit (2013, 03). Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics* 14(3), 586–599.
- Alhamzawi, R. and H. T. M. Ali (2018). The Bayesian adaptive lasso regression. *Mathematical Biosciences* 303, 75–82.
- Bai, R. (2020). Bayesian group regularization in generalized linear models with a continuous spike-and-slab prior. *arXiv preprint arXiv:2007.07021*.
- Bai, R., G. E. Moran, J. L. Antonelli, Y. Chen, and M. R. Boland (2022). Spike-and-slab group lassos for grouped regression and sparse generalized additive models. *Journal of the American Statistical Association* 117(537), 184–197.
- Bai, R., V. Ročková, and E. I. George (2021). Spike-and-slab meets LASSO: A review of the spike-and-slab LASSO. In *Handbook of Bayesian Variable Selection*, pp. 81–108. Chapman and Hall/CRC.
- Bartoň, M. and V. M. Calo (2016). Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 305, 217–240.
- Bayarri, M. J., J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu (2007). A framework for validation of computer models. *Technometrics* 49(2), 138–154.
- Beal, M. J., F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild (2004, 09). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics* 21(3), 349–356.
- Berk, R., L. Brown, A. Buja, K. Zhang, and L. Zhao (2013). Valid post-selection inference. *The Annals of Statistics* 41(2), 802–837.

- Berry, S. M., R. J. Carroll, and D. Ruppert (2002). Bayesian smoothing and regression splines for measurement error problems. *Journal of the American Statistical Association* 97(457), 160–169.
- Breheny, P. (2015). The group exponential lasso for bi-level variable selection. *Biometrics* 71, 731–740.
- Calderhead, B., M. Girolami, and N. Lawrence (2008). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. *Advances in Neural Information Processing Systems* 21, 217–224.
- Campbell, D. A. (2007). *Bayesian collocation tempering and generalized profiling for estimation of parameters from differential equation models*. Ph. D. thesis, McGill University.
- Chen, S., A. Shojaie, and D. M. Witten (2017). Network reconstruction from high-dimensional ordinary differential equations. *Journal of the American Statistical Association* 112(520), 1697–1707.
- Dai, X. and L. Li (2022). Kernel ordinary differential equations. *Journal of the American Statistical Association* 117(540), 1711–1725.
- Dai, X. and L. Li (2024). Post-regularization confidence bands for ordinary differential equations. *Journal of Machine Learning Research* 25(23), 1–51.
- Dattner, I., S. Gugushvili, and O. Laskorunskyi (2024). Model selection for ordinary differential equations: A statistical testing approach. *Biometrical Journal* 66(8), e70013.
- Dattner, I. and C. A. J. Klaassen (2015). Optimal rate of direct estimators in systems of ordinary differential equations linear in functions of the parameters. *Electronic Journal of Statistics* 9(2), 1939 – 1973.
- de Boor, C., T. Lyche, and L. L. Schumaker (1976). On calculating with B-splines II. Integration. *Numerische Methoden der Approximationstheorie/Numerical Methods of Approximation Theory*, 123–146.
- Dondelinger, F., D. Husmeier, S. Rogers, and M. Filippone (2013). ODE parameter inference using adaptive gradient matching with Gaussian processes. In *Artificial Intelligence and Statistics*, pp. 216–228. PMLR.
- Duarte, C. W. and Z.-B. Zeng (2011). High-confidence discovery of genetic network regulators in expression quantitative trait loci data. *Genetics* 187(3), 955–964.

- Fan, J. and I. Gijbels (1995). *Local Polynomial Modelling and Its Applications*. CRC, London.
- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal* 1(6), 445–466.
- Gause, G. F. (1935). Experimental demonstration of Volterra’s periodic oscillations in the numbers of animals. *Journal of Experimental Biology* 12(1), 44–48.
- Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis* 1(3), 515 – 534.
- Golub, G. H. and J. H. Welsch (1969). Calculation of Gauss quadrature rules. *Mathematics of Computation* 23(106), 221–230.
- Gronau, Q. F., H. Singmann, and E.-J. Wagenmakers (2020). bridgesampling: An R package for estimating normalizing constants. *Journal of Statistical Software* 92(10), 1–29.
- Gupta, M., P. Qu, and J. G. Ibrahim (2007). A temporal hidden Markov regression model for the analysis of gene regulatory networks. *Biostatistics* 8(4), 805–820.
- Hausser, J. and K. Strimmer (2009). Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research* 10(7), 1469–1484.
- Hewitt, C. G. (1921). *The Conservation of the Wild Life of Canada*. New York: C. Scribner.
- Homan, M. D. and A. Gelman (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* 15(1), 1593–1623.
- Hooker, G., J. O. Ramsay, and L. Xiao (2016). CollocInfer: Collocation inference in differential equation models. *Journal of Statistical Software* 75(2), 1–52.
- Hsu, J. C., J. Y. Chang, and T. Wang (2006). Simultaneous confidence intervals for differential gene expressions. *Journal of Statistical Planning and Inference* 136(7), 2182–2196. In Memory of Dr. Shanti Swarup Gupta.
- Huang, H., A. Handel, and X. Song (2020). A Bayesian approach to estimate parameters of ordinary differential equation. *Computational Statistics* 35(3), 1481–1499.

- Kang, J. and J. Guo (2009). Self-adaptive Lasso and its Bayesian estimation. Technical report, Working Paper.
- Karlebach, G. and R. Shamir (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology* 9(10), 770–780.
- Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association* 90(430), 773–795.
- Kennedy, M. C. and A. O’Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63(3), 425–464.
- Kim, J. (2016). Validation and selection of ODE models for gene regulatory networks. *Chemometrics and Intelligent Laboratory Systems* 157, 104–110.
- Kim, J., J. Li, S. G. Venkatesh, D. S. Darling, and G. A. Rempala (2013). Model discrimination in dynamic molecular systems: Application to parotid de-differentiation network. *Journal of Computational Biology* 20(7), 524–539.
- Kim, Y. and J. Kim (2019). Model validation in dynamic systems for time-course data with complex error structures. *Journal of Chemometrics* 33(3), e3108. e3108 cem.3108.
- Kristensen, K., A. Nielsen, C. W. Berg, H. Skaug, and B. M. Bell (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software* 70(5), 1–21.
- Lam, F. S., C. C. Yip, Z. A. Lee, and P. H. Ong (2023). Analyzing post-disaster reconstruction stakeholder networks: Malaysian rural housing. *Environment-Behaviour Proceedings Journal* 8(23), 147–155.
- Leslie, P. H. (1957). An analysis of the data for some experiments carried out by Gause with populations of the protozoa, *Paramecium aurelia* and *Paramecium caudatum*. *Biometrika* 44(3/4), 314–327.
- Liang, F., Q. Song, and P. Qiu (2015). An equivalent measure of partial correlation coefficients for high-dimensional Gaussian graphical models. *Journal of the American Statistical Association* 110(511), 1248–1265.
- Liang, H. and H. Wu (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association* 103(484), 1570–1583.

- Liu, J., H. Kang, M. Raab, A. J. Da Silva, S.-K. Kraeft, and C. E. Rudd (1998). FYB (FYN binding protein) serves as a binding partner for lymphoid protein and FYN kinase substrate SKAP55 and a SKAP55-related protein in T cells. *Proceedings of the National Academy of Sciences* 95(15), 8779–8784.
- Liu, Z. and M. Li (2022). Optimal plug-in Gaussian processes for modelling derivatives. *arXiv preprint arXiv:2210.11626*.
- Lotka, A. J. (1932). The growth of mixed populations: Two species competing for a common food supply. *Journal of the Washington Academy of Sciences* 22(16/17), 461–469.
- Lu, T., H. Liang, H. Li, and H. Wu (2011). High-dimensional ODEs coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association* 106(496), 1242–1258.
- Mandric, I., Y. Temate-Tiagueu, T. Shcheglova, S. Al Seesi, A. Zelikovsky, and I. I. Măndoiu (2017, 06). Fast bootstrapping-based estimation of confidence intervals of expression levels and differential expression from RNA-Seq data. *Bioinformatics* 33(20), 3302–3304.
- Marlin, T. E. Process control, designing processes and control systems for dynamic performance.
- May, R. M. (1973). *Stability and Complexity in Model Ecosystems*. Princeton University Press.
- Monnahan, C. C. and K. Kristensen (2018). No-U-turn sampling for fast Bayesian inference in ADMB and TMB: Introducing the adnuts and tmbstan R packages. *PLOS ONE* 13(5), 1–10.
- Murdoch, W. W., C. J. Briggs, and R. M. Nisbet (2013). *Consumer-resource dynamics*. Princeton University Press.
- Nanshan, M., N. Zhang, X. Xun, and J. Cao (2022). Dynamical modeling for non-Gaussian data with high-dimensional sparse ordinary differential equations. *Computational Statistics & Data Analysis* 173, 107483.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*, pp. 113–162. Chapman and Hall/CRC.

- Nikolov, G. (1996). On certain definite quadrature formulae. *Journal of Computational and Applied Mathematics* 75(2), 329–343.
- Odum, E. P. (1953). *Fundamentals of Ecology*. Saunders Philadelphia.
- Poyton, A., M. S. Varziri, K. B. McAuley, P. J. McLellan, and J. O. Ramsay (2006). Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers & Chemical Engineering* 30(4), 698–708.
- Pringsheim, E. G. (1928). Physiologische untersuchungen an paramaecium bursaria: ein beitrag zur symbioseforschung. *Arch. f. Protistenk* 64, 289–418.
- Qi, X. and H. Zhao (2010). Asymptotic efficiency and finite-sample properties of the generalized profiling estimation of parameters in ordinary differential equations. *The Annals of Statistics* 38(1), 435 – 481.
- Raab, M., H. Kang, A. da Silva, X. Zhu, and C. E. Rudd (1999). FYN-T-FYB-SLP-76 interactions define a T-cell receptor ζ /CD3-mediated tyrosine phosphorylation pathway that up-regulates interleukin 2 transcription in T-cells. *Journal of Biological Chemistry* 274(30), 21170–21179.
- Ramsay, J. O., G. Hooker, D. Campbell, and J. Cao (2007). Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69(5), 741–796.
- Ramsay, J. O. and B. W. Silverman (2008). Functional data analysis. *Internet Adresi: http*.
- Rangel, C., J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani (2004). Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics* 20(9), 1361–1372.
- Rangel, C., J. Angus, Z. Ghahramani, and D. L. Wild (2005). Modeling genetic regulatory networks using gene expression profiling and state-space models. *Probabilistic Modeling in Bioinformatics and Medical Informatics*, 269–293.
- Rempala, G., K. Ramos, T. Kalbfleisch, and I. Teneng (2007, 05). Validation of a mathematical model of gene transcription in aggregated cellular systems: Application to 11 retrotransposition. *Journal of computational biology : a journal of computational molecular cell biology* 14, 339–49.

- Soetaert, K., T. Petzoldt, and R. W. Setzer (2010). Solving differential equations in R: Package deSolve. *Journal of Statistical Software* 33(9), 1–25.
- St John, P. C. and F. J. Doyle III (2013). Estimating confidence intervals in predicted responses for oscillatory biological models. *BMC systems biology* 7(1), 71.
- Stan Development Team (2022). RStan: the R interface to Stan. R package version 2.21.7.
- Stone, C. J. (1985). Additive regression and other nonparametric models. *The Annals of Statistics* 13(2), 689–705.
- Stroud, A. H. and D. Secrest (1966). *Gaussian Quadrature Formulas*. Prentice-Hall.
- Takagane, K., M. Umakoshi, G. Itoh, S. Kuriyama, A. Goto, and M. Tanaka (2022). SKAP2 suppresses inflammation-mediated tumorigenesis by regulating SHP-1 and SHP-2. *Oncogene* 41(8), 1087–1099.
- Tan, J., G. Zhang, X. Wang, H. Huang, and F. Yao (2024, 04). Green’s matching: an efficient approach to parameter estimation in complex dynamic systems. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 86(5), 1266–1285.
- Tronarp, F., N. Bosch, and P. Hennig (2022). Fenrir: Physics-enhanced regression for initial value problems. In *International Conference on Machine Learning*, pp. 21776–21794. PMLR.
- Varah, J. M. (1982). A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing* 3(1), 28–46.
- Vyshemirsky, V. and M. A. Girolami (2008, 3). Bayesian ranking of biochemical system models. *Bioinformatics* 24(6), 833–839.
- Wang, H. and C. Leng (2008). A note on adaptive group Lasso. *Computational Statistics & Data Analysis* 52(12), 5277–5286.
- Wang, S., S. Ge, R. Doig, and L. Wang (2022). Adaptive semiparametric Bayesian differential equations via sequential Monte Carlo. *Journal of Computational and Graphical Statistics* 31(2), 600–613.
- Wang, Z. and M. Peng (2021). A novel prognostic biomarker LCP2 correlates with metastatic melanoma-infiltrating CD8+ T cells. *Scientific reports* 11(1), 9164.

- Wenk, P., A. Gotovos, S. Bauer, N. S. Gorbach, A. Krause, and J. M. Buhmann (2019). Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1351–1360. PMLR.
- Wu, H., T. Lu, H. Xue, and H. Liang (2014). Sparse additive ordinary differential equations for dynamic gene regulatory network modeling. *Journal of the American Statistical Association* 109(506), 700–716.
- Wu, L., X. Qiu, Y.-x. Yuan, and H. Wu (2019). Parameter estimation and variable selection for big systems of linear ordinary differential equations: A matrix-based approach. *Journal of the American Statistical Association* 114(526), 657–667.
- Wu, M. and M. Lysy (2024, 02–04 May). Data-adaptive probabilistic likelihood approximation for ordinary differential equations. In S. Dasgupta, S. Mandt, and Y. Li (Eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, Volume 238 of *Proceedings of Machine Learning Research*, pp. 1018–1026. PMLR.
- Wu, X., J. Thisdelle, S. Hou, J. E. Fajardo-Despaigne, S. B. Gibson, J. B. Johnston, D. E. Dawe, V. Banerji, and A. J. Marshall (2023). Elevated expression of interleukin 16 in chronic lymphocytic leukemia is associated with disease burden and abnormal immune microenvironment. *Leukemia Research* 131, 107315.
- Xu, M., S. W. K. Wong, and P. Sang (2024). A Bayesian collocation integral method for parameter estimation in ordinary differential equations. *Journal of Computational and Graphical Statistics* 33(3), 845–854.
- Xu, X. and M. Ghosh (2015). Bayesian variable selection and estimation for group lasso. *Bayesian Analysis* 10(4), 909 – 936.
- Yang, S., S. W. K. Wong, and S. C. Kou (2021). Inference of dynamic systems from noisy and sparse data via manifold-constrained Gaussian processes. *Proceedings of the National Academy of Sciences* 118(15), e2020397118.
- Zhang, N., M. Nanshan, and J. Cao (2022). A joint estimation approach to sparse additive ordinary differential equations. *Statistics and Computing* 32(5), 69.
- Zhang, T., J. Wu, F. Li, B. Caffo, and D. Boatman-Reich (2015). A dynamic directional model for effective brain connectivity using electrocorticographic (ECoG) time series. *Journal of the American Statistical Association* 110(509), 93–106.

APPENDICES

Appendix A

Implementation Details

A.1 Details of sampling algorithm

We provide additional details on how Markov chain Monte Carlo (MCMC) samples are drawn from the target posterior distribution associated with our proposed method. As mentioned in the main text, the no-U-turn sampler (NUTS, [Homan and Gelman, 2014](#)) is used as the MCMC sampling algorithm. A standard Hamiltonian Monte Carlo (HMC, [Neal, 2011](#)) algorithm can be sensitive to the step size parameter ϵ and the number of steps L . In contrast, NUTS, as an extension of HMC, automatically controls the parameter L and has been shown to perform at least as efficiently as a well-tuned HMC algorithm.

As in the setup for HMC, suppose the target distribution has density $\pi_{\text{target}}(\mathbf{q}) = (1/Z) \exp\{\mathcal{L}(\mathbf{q})\}$, where Z denotes the normalizing constant and $\mathcal{L}(\mathbf{q})$ is the logarithm of the joint density of the variable of interest $\mathbf{q}$ (in vector form). Thus in our proposed method, we treat $\mathbf{q} \equiv (\boldsymbol{\theta}, \mathbf{c}, \boldsymbol{\sigma})$, so $\mathcal{L}(\mathbf{q})$ corresponds to the logarithm of the posterior density (8) in the main text.

Each NUTS iteration then proceeds as follows. First, we sample a momentum variable $\mathbf{r}$ from the multivariate normal distribution $MVN(0, \mathbf{I}_{|\mathbf{q}| \times |\mathbf{q}|})$, i.e., $\mathbf{r}$ has the same dimension as $\mathbf{q}$. Second, we sample a slice variable $\mathbf{u}$ from a Uniform $([0, \exp\{\mathcal{L}(\mathbf{q}) - \frac{1}{2}\mathbf{r}^\top \mathbf{r}\}])$ distribution; the term $\frac{1}{2}\mathbf{r}^\top \mathbf{r}$ is known as the kinetic energy. Given the sampled $\mathbf{r}$ and $\mathbf{u}$, a leapfrog integrator is introduced to trace out a path for $(\mathbf{r}, \mathbf{q})$: first we move forwards or backwards 1 leapfrog step, then forwards or backwards 2 leapfrog steps, then forwards or backwards 4 leapfrog steps, etc. This doubling process implicitly constructs a balanced binary tree with leaf nodes representing the states of $(\mathbf{r}, \mathbf{q})$. This process is repeated until

the so-called U-turn condition is met, i.e., the path between the leftmost and rightmost nodes of any balanced subtree starts doubling back on itself, or the path encounters a state with very low probability. Finally, we sample the new state $(\mathbf{r}^*, \mathbf{q}^*)$ uniformly at random from the leapfrog path that lies within the slice defined by the slice variable $\mathbf{u}$; this procedure yields the next MCMC draw and satisfies detailed balance.

A.2 Software Setup

For Chapter 2 where ODEs have known function forms of $\mathbf{f}$, we implement the posterior distribution (2.5) in C++ and pass it to the *tmbstan* package (Monnahan and Kristensen, 2018) as described in Section 2.2.4 of the main text. For the systems analyzed in our simulation studies, we set the maximum tree depth of the binary tree to 10 and the target average proposal acceptance probability to 0.85. The code for reproducing the simulation studies and real data analysis can be found in the supplementary material of the paper “A Bayesian Collocation Integral Method for Parameter Estimation in Ordinary Differential Equations” published in the *Journal of Computational and Graphical Statistics* (Xu et al., 2024).

Similarly, for Chapters 3 and 4, the corresponding posterior distributions (3.11) and (4.6) are implemented in C++ and interfaced with the *tmbstan* package, as described in Sections 3.2.4 and 4.2.4 of the main text, respectively. For all simulated ODE systems analyzed, we set the maximum tree depth of the binary tree to 12 and the target average proposal acceptance probability to 0.85.

Appendix B

Additional Results for Chapter 2

B.1 Simulation studies: additional results

B.1.1 FitzHugh-Nagumo (FN) system

To better assess the robustness of the estimation, we present the boxplots for the estimated parameter and trajectory RMSEs across 100 simulation runs under different sampling frequencies. We use the formula given in Section 2.3.2 of the main text to calculate trajectory RMSEs. To calculate parameter RMSEs (e.g., in Table 1 of the main text), if we let the parameter estimate of θ for the j -th simulation repetition be denoted as $\hat{\theta}_j$, then the parameter RMSE (over the 100 simulations) is calculated as

$$\sqrt{\frac{1}{100} \sum_{j=1}^{100} (\hat{\theta}_j - \theta)^2} ;$$

while in the corresponding boxplot (e.g., in Figures B.1, B.2 and B.3), the parameter RMSE for each individual simulation run is simply $|\hat{\theta}_j - \theta|$, and these 100 parameter RMSEs are plotted.

Figures B.1, B.2 and B.3 display the boxplots of parameter RMSEs under the settings of 41, 21 and 11 noisy observations, respectively. For the trajectory RMSEs, Figures B.4, B.5 and B.6 depict the boxplots under the settings of 41, 21 and 11 noisy observations, respectively. With more dense observations, the parameter and trajectory RMSEs will certainly be lower for each method considered; however, it is in the intermediate case

(i.e., 21 observations in the FN system) where the better performance of our method is overall most evident, when compared to the other methods in terms of trajectory RMSEs. When observations are dense enough to generate a reliable estimate for the derivatives, the differences between methods are relatively minor, but the integral-based method is still superior. As the observations become sparser, the derivatives cannot be estimated that accurately; consequently, the advantage of the integral-based method becomes particularly evident in recovering the trajectories. With extremely sparse observations, recovering the system accurately is challenging for any method; the integral method is still superior but the differences are less pronounced. As shown in Figures B.4, B.5 and B.6, our proposed method generally outperforms other comparison methods in terms of trajectory RMSEs more evidently for 21 observations than 41 or 11 observations. Further discussion of these figures is provided in Section 2.3.3 of the main text.

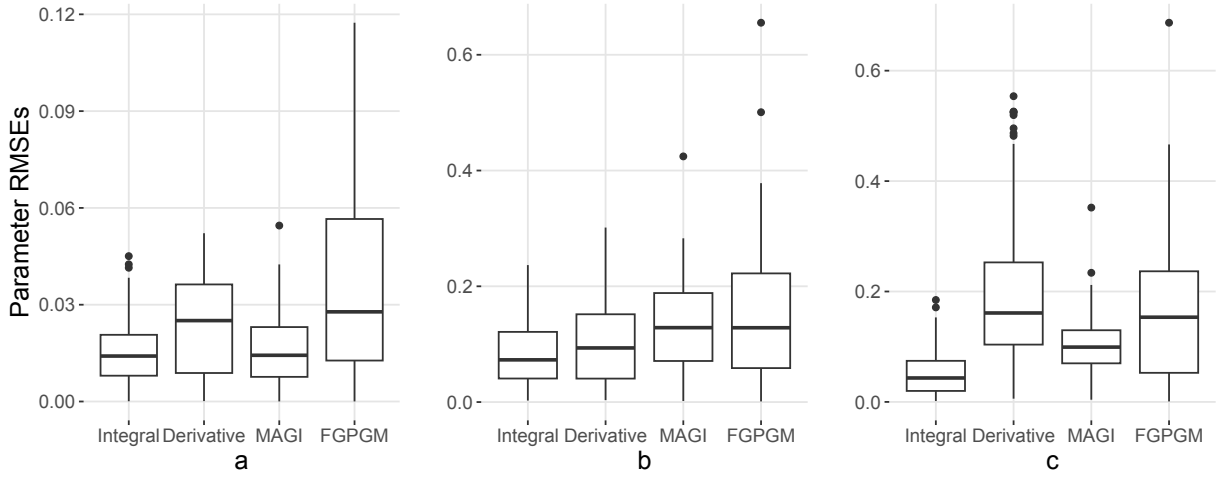


Figure B.1: Estimated parameter RMSEs for a , b and c respectively in the FN system, over 100 simulation runs with 41 observations.

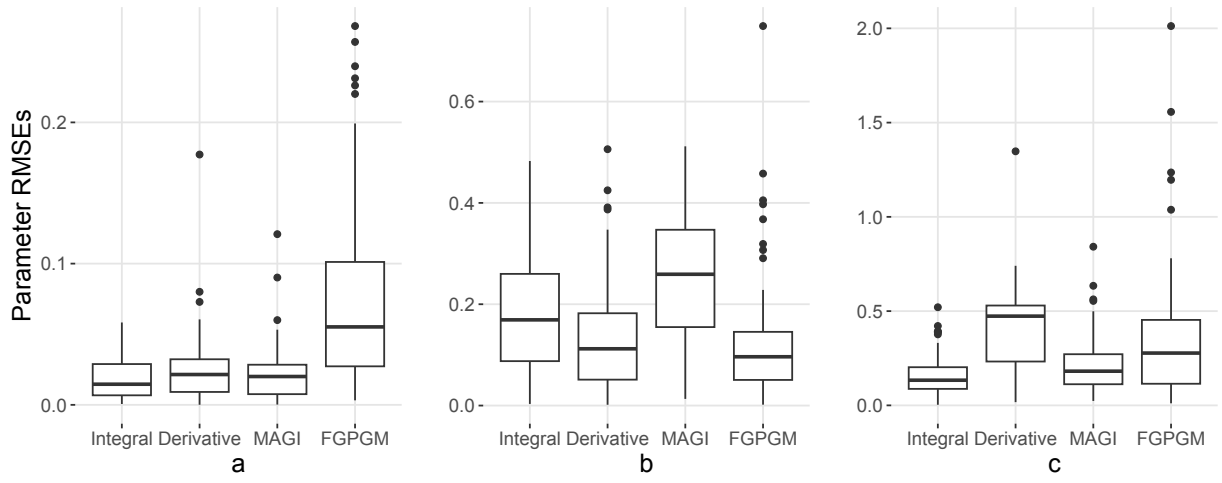


Figure B.2: Estimated parameter RMSEs for a , b and c respectively in the FN system, over 100 simulation runs with 21 observations.

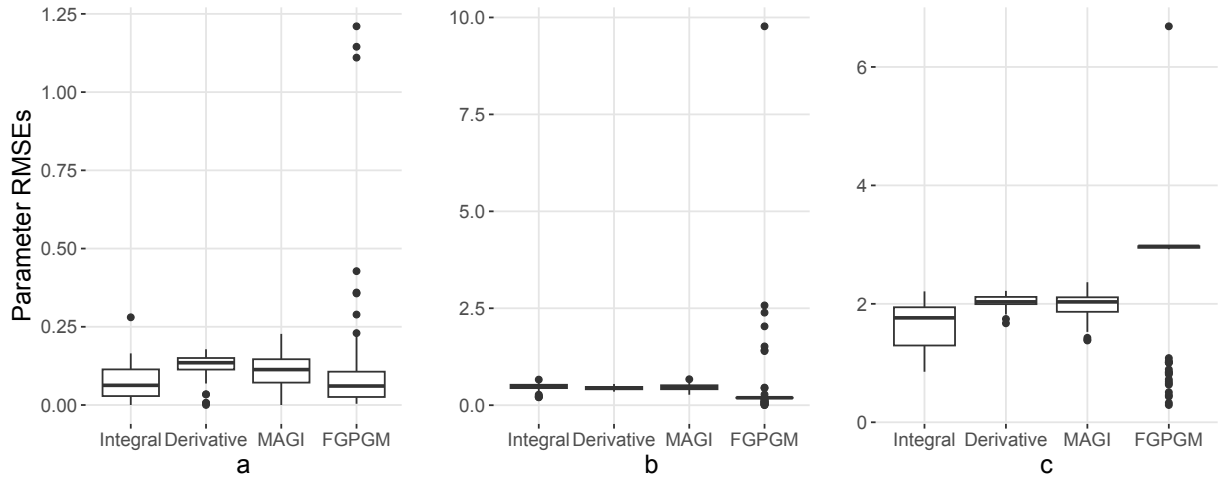


Figure B.3: Estimated parameter RMSEs for a , b and c respectively in the FN system, over 100 simulation runs with 11 observations.

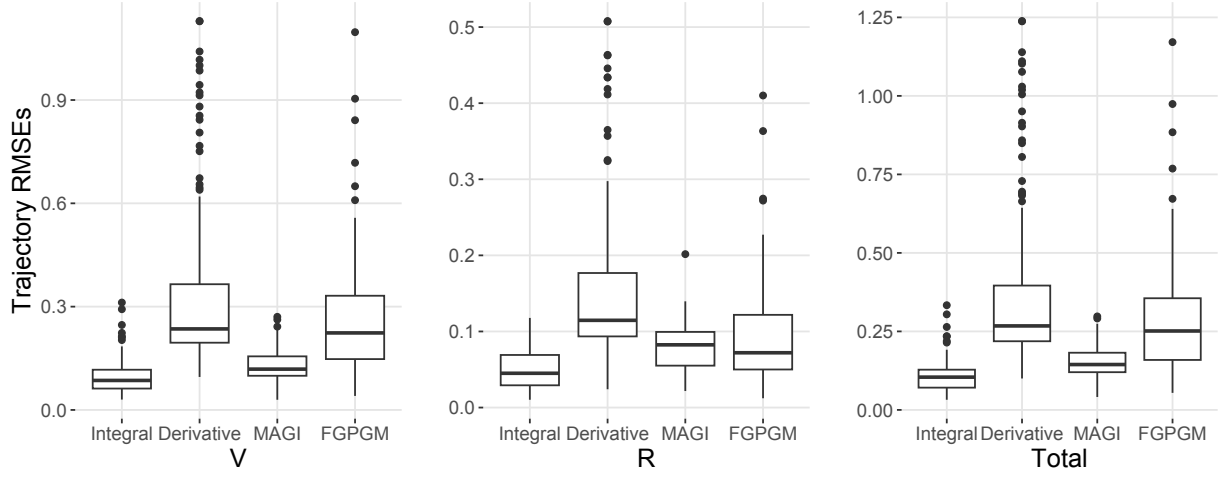


Figure B.4: Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 41 observations.

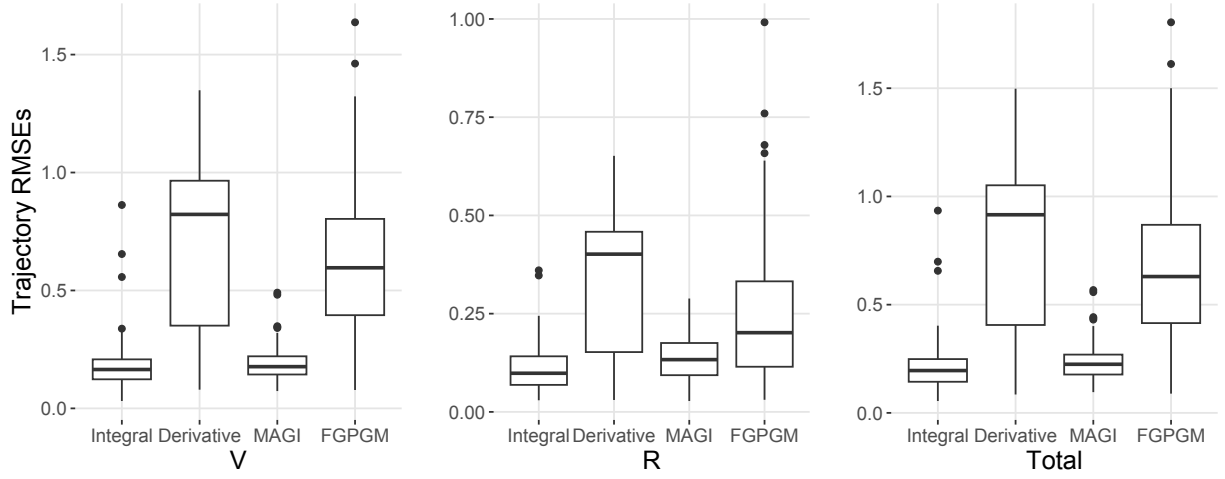


Figure B.5: Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 21 observations.

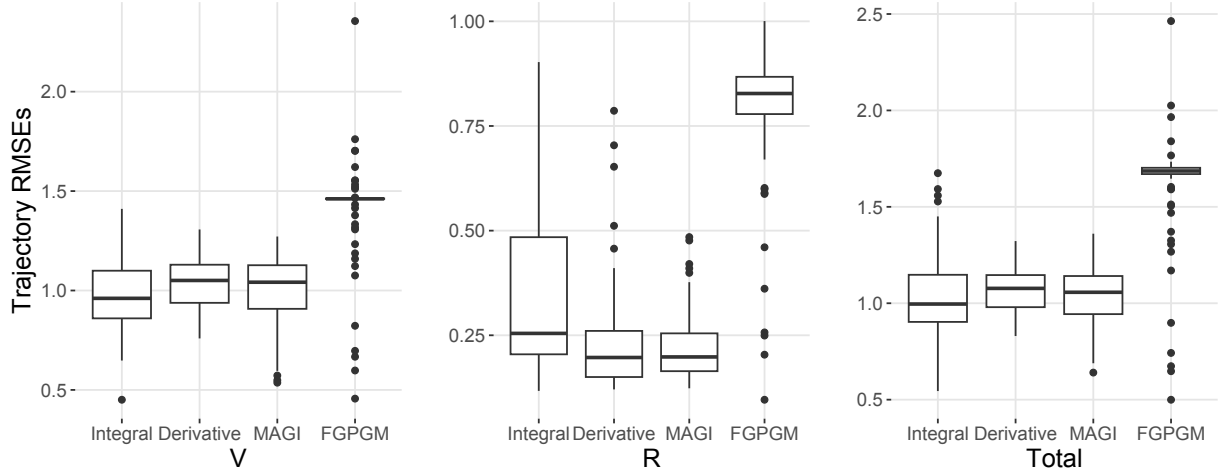


Figure B.6: Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 11 observations.

To better study the performance of our method in reconstructing the trajectories, we report the estimated norm of each component. Table B.1 displays the average norm of component V and R across the 100 simulation runs under different sampling frequencies. For each method, the norm of each estimated component is generally closer to that of the true one for the system when estimated with denser observations than with sparser observations. Moreover, the discrepancy in the norm between the estimated and true components is more evident for V than R , as V is more bumpy than R . For each sampling frequency, the norms of the estimated components obtained from our proposed method are the closest relative to those of the true components among all methods, except component R in the case of 11 observations and component V in the 21 observations case. In particular, for component V , this difference becomes even more apparent under the setting of sparser observations. Thus, we can draw the same conclusion as in the main text that our proposed method is more efficient in recovering trajectories regardless of sampling frequency, and the advantage is more obvious when recovering a wiggly component from sparser observations.

Observations	Method	V	R
41	True	47.079	21.972
	Integral	46.631	21.848
	Derivative	46.334	22.134
	MAGI	46.215	21.806
	FGPGM	46.064	22.000
21	Integral	45.953	21.892
	Derivative	45.955	22.715
	MAGI	45.333	21.766
	FGPGM	46.603	22.935
11	Integral	30.589	24.318
	Derivative	21.699	23.426
	MAGI	23.191	23.142
	FGPGM	16.718	15.767

Table B.1: Average of the norms of each estimated component in the FN system across 100 simulation runs.

To further gauge the performance of our method in estimating the ODE solution, we plot the estimated trajectories for one particular simulation run. From the 100 simulation runs, one random dataset is selected at each sampling frequency, as shown in Figures B.7, B.8, and B.9. As the observations become sparser, the discrepancy between the estimated and true trajectories becomes more pronounced, and the 95% pointwise credible intervals become wider. Nevertheless, all estimated trajectories provide reasonable fits, and all 95% pointwise credible intervals cover the true trajectories.

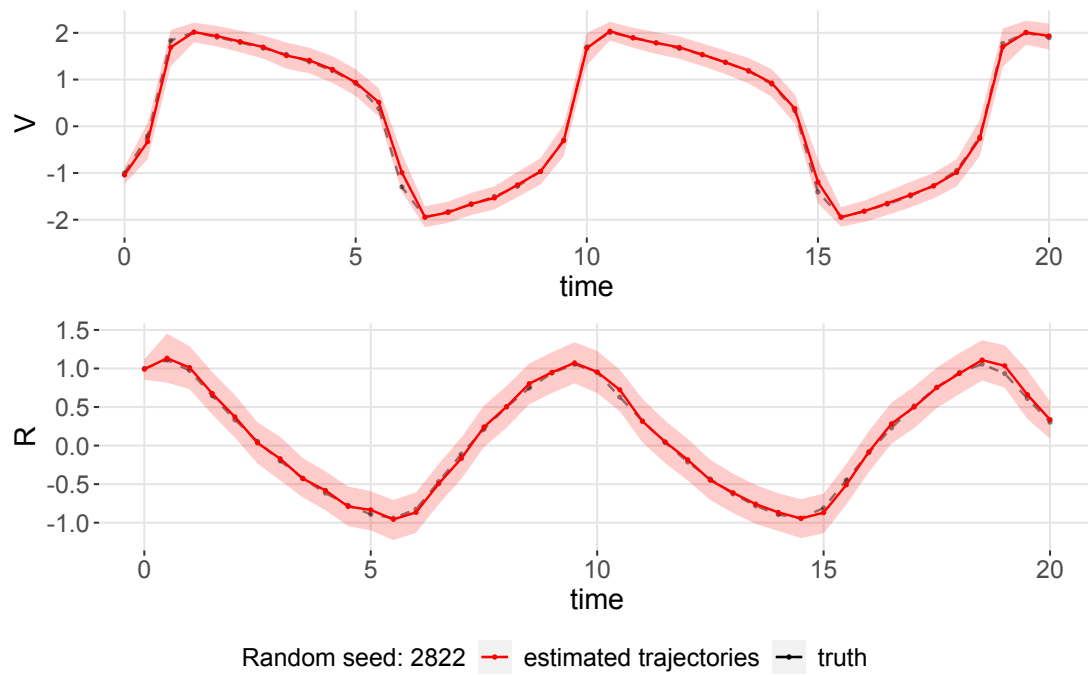


Figure B.7: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations of one random simulation run. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

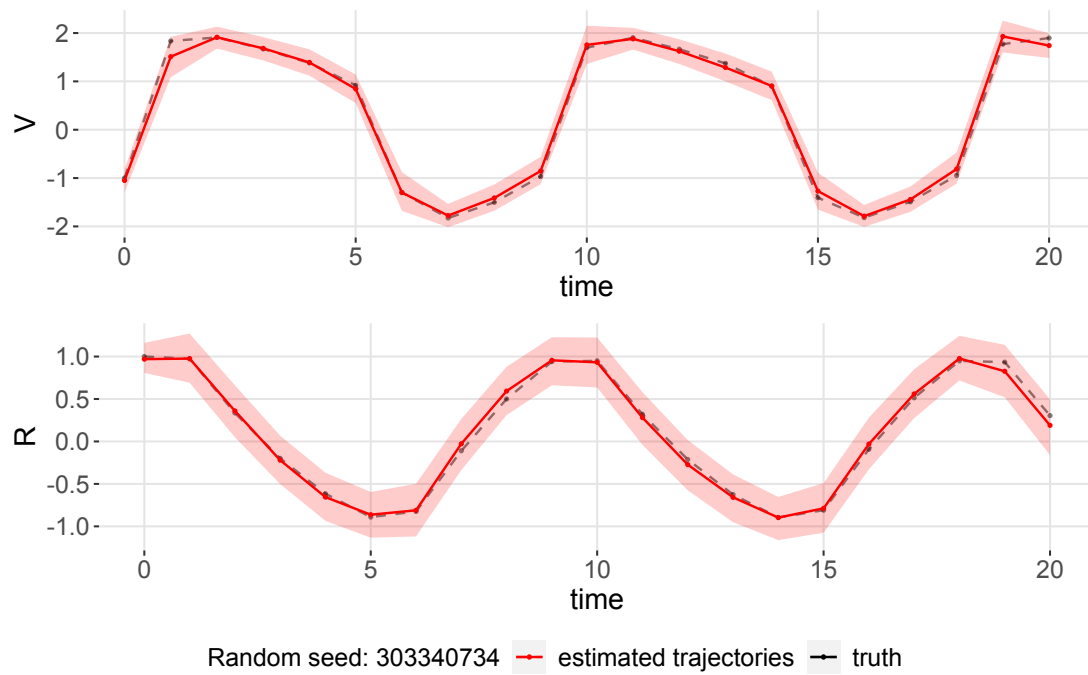


Figure B.8: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 21 observations of one random simulation run. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

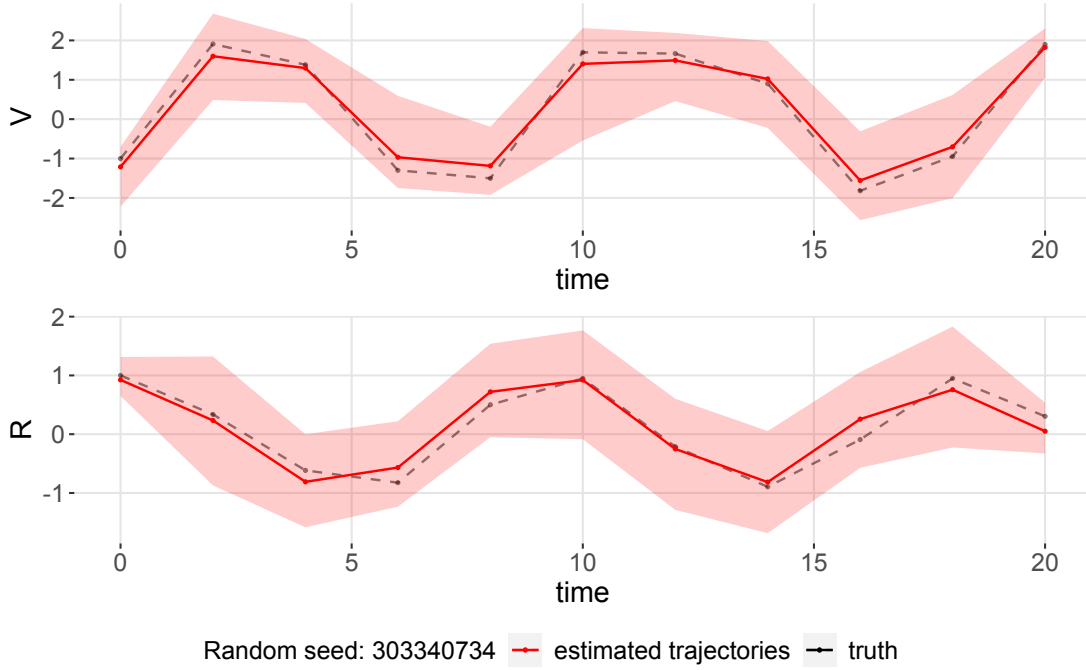


Figure B.9: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 11 observations of one random simulation run. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

Table B.2 presents the average running time (in minutes) of implementing each method for the FN system under different sampling frequencies. Since our integral method requires numerical approximations for the integrals, it has a higher computational cost than the derivative method. However, in the case of 41 observations, our method is about 1.5 times faster than MAGI and 62 times faster than FGPGM on the same processor. For sparser cases, the computational cost of our method is lower than that for the dense case, while MAGI requires more computational time.

Observations	Integral	Derivative	MAGI	FGPGM
41	3.28	0.17	5.23	203.65
21	3.05	0.25	8.55	82.35
11	0.54	0.05	8.63	41.61

Table B.2: Average running time (minutes) of each method for the FN system across 100 simulation runs.

B.1.2 Protein transduction system

We now consider a more challenging system with unidentifiable parameters: the protein transduction system (Vysheirsky and Girolami, 2008). This artificial biochemical system consists of five components: through the biochemical reactions, input signal protein S can degrade to S_d ; at the same time, S can bind to protein R to form the complex S_R and activate R to its phosphorylated form R_{pp} . Then $\mathbf{x} = (S, S_d, R, S_R, R_{pp})$ satisfies the ODE

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}, t) = \begin{bmatrix} -k_1 \cdot S - k_2 \cdot S \cdot R + k_3 \cdot S_R \\ k_1 \cdot S \\ -k_2 \cdot S \cdot R + k_3 \cdot S_R + \frac{V \cdot R_{pp}}{K_m + R_{pp}} \\ k_2 \cdot S \cdot R - k_3 \cdot S_R - k_4 \cdot S_R \\ k_4 \cdot S_R - \frac{V \cdot R_{pp}}{K_m + R_{pp}} \end{bmatrix}, \quad (\text{B.1})$$

where the parameter vector is $\boldsymbol{\theta} = (k_1, k_2, k_3, k_4, V, K_m)$. Following the same setup as in Yang et al. (2021), we set the observation times as $t = \{0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100\}$, so that the observations are unequally-spaced and relatively sparse within the time interval. The true parameter values are set as $\boldsymbol{\theta} = (0.07, 0.6, 0.05, 0.3, 0.017, 0.3)$ and the initial conditions are set as $\mathbf{x}(0) = (1, 0, 1, 0, 0)$. Noisy observations are generated with two noise levels for all components: $\sigma = 0.001$ (low noise) and $\sigma = 0.01$ (high noise). Following Yang et al. (2021), we set the priors for $\boldsymbol{\theta}$ uniformly distributed on $[0, 4]$. For both noise settings, we place 15 knots only at each observed time point with 17 basis functions, and the smoothing parameter selection algorithm starts from $\lambda^{(0)} = 10^2$ with $\lambda^* = 10^3$. Following the guidelines in Section 2.2.2 of the main text, $M_i = 60$ quadrature points are chosen for estimating the outer integral. Since the system involves a polynomial of degree 2, we use $K_{m_i} = 4$ quadrature points for the inner integral.

Similar to the FN system, we compare our proposed method with three methods: MAGI (Yang et al., 2021), FGPGM (Wen et al., 2019) and the Bayesian collocation method that replaces prior (2.4) containing only integrals with prior (2.3) containing derivatives. All

comparison methods use the same prior θ as the proposed method, which is uniform on $[0, 4]$. To implement these two GP-based methods, we strictly follow the recommendation of the authors. In particular, we implement MAGI using their R package *magi*: first, 101 equally-spaced discretization points are used to optimize hyperparameter $\hat{\sigma}$; second, with the estimated $\hat{\sigma}$, same discretizations are used to obtain the optimized hyperparameters $(\tilde{\phi}, \tilde{\sigma})$; finally, we use 201 discretizations and optimized $(\tilde{\phi}, \tilde{\sigma})$ to obtain the finalized parameter estimates. For each optimization and parameter estimation step, 20,000 HMC iterations are run, including a 10,000-iteration warmup. Moreover, each HMC iteration uses 100 leapfrog steps. For FGPGM, we run their software: the standard deviation γ that adjusts the potential model mismatch between the ODE output and GP derivatives is set to 10^{-4} , along with a sigmoid kernel. 300,000 MCMC iterations are run, where the first half of the iterations is treated as a warmup. The Bayesian collocation with derivative estimation takes the exact same setting as our method described in Section 2.2.4. Because of the nonidentifiability for parameters V and K_m , the RMSEs of parameter estimation are not meaningful for this system. Thus similar to previous authors who studied this system (Wenk et al., 2019; Yang et al., 2021), we focus on trajectory RMSEs. For each of these comparison methods, the posterior means are used as parameter estimates for only calculating the trajectory RMSEs. A total of 100 independent simulation trials are run. In all simulation studies, the R package *deSolve* (Soetaert et al., 2010) is used for numerically solving the differential equations.

Table B.3 summarizes the results of the median and the interquartile range (IQR) of trajectory RMSEs for the protein transduction system under different noise settings, and Figure B.10 depicts the boxplot for trajectory RMSEs for the four methods over 100 simulation runs under the low noise setting. Under the low noise setting, our proposed method has the lowest trajectory RMSEs for each component, while the trajectory RMSE of the derivative method is higher than that of MAGI. For components S_d , R and R_{pp} , the derivative method has even higher RMSEs than FGPGM. Similar to the FN system, we plot the average of the estimated trajectories across 100 simulation runs as well as the 95% pointwise intervals, as shown in Figure B.11. Our method can accurately recover the system under the low noise setting, and the 95% pointwise credible intervals are quite narrow.

Noise level	Method	S	S_d	R	S_R	R_{pp}	Total
Low-noise, $\sigma = 0.001$	Integral	0.0003	0.0004	0.0013	0.0003	0.0013	0.0020
		(0.0003)	(0.0005)	(0.0017)	(0.0002)	(0.0017)	(0.0022)
	Derivative	0.0007	0.0059	0.0250	0.0011	0.0255	0.0372
		(0.0009)	(0.0066)	(0.0045)	(0.0010)	(0.0041)	(0.0062)
	MAGI	0.0009	0.0015	0.0049	0.0008	0.0051	0.0073
		(0.0001)	(0.0008)	(0.0018)	(0.0001)	(0.0015)	(0.0019)
	FGPGM	0.0021	0.0013	0.0213	0.0016	0.0214	0.0304
		(0.0002)	(0.0015)	(0.0027)	(0.0002)	(0.0028)	(0.0040)
High-noise, $\sigma = 0.01$	Integral	0.0016	0.0025	0.0051	0.0015	0.0052	0.0085
		(0.0011)	(0.0020)	(0.0036)	(0.0009)	(0.0034)	(0.0049)
	Derivative	0.0021	0.0085	0.0290	0.0027	0.0299	0.0443
		(0.0020)	(0.0109)	(0.0172)	(0.0019)	(0.0109)	(0.0194)
	MAGI	0.0042	0.0034	0.0165	0.0045	0.0184	0.0264
		(0.0015)	(0.0036)	(0.0038)	(0.0024)	(0.0042)	(0.0042)
	FGPGM	0.0053	0.0071	0.0215	0.0057	0.0371	0.0472
		(0.0030)	(0.0080)	(0.0098)	(0.0039)	(0.0209)	(0.0182)

Table B.3: Median of trajectory RMSEs (with IQR in parentheses) for each component of the protein transduction system across 100 simulation runs.

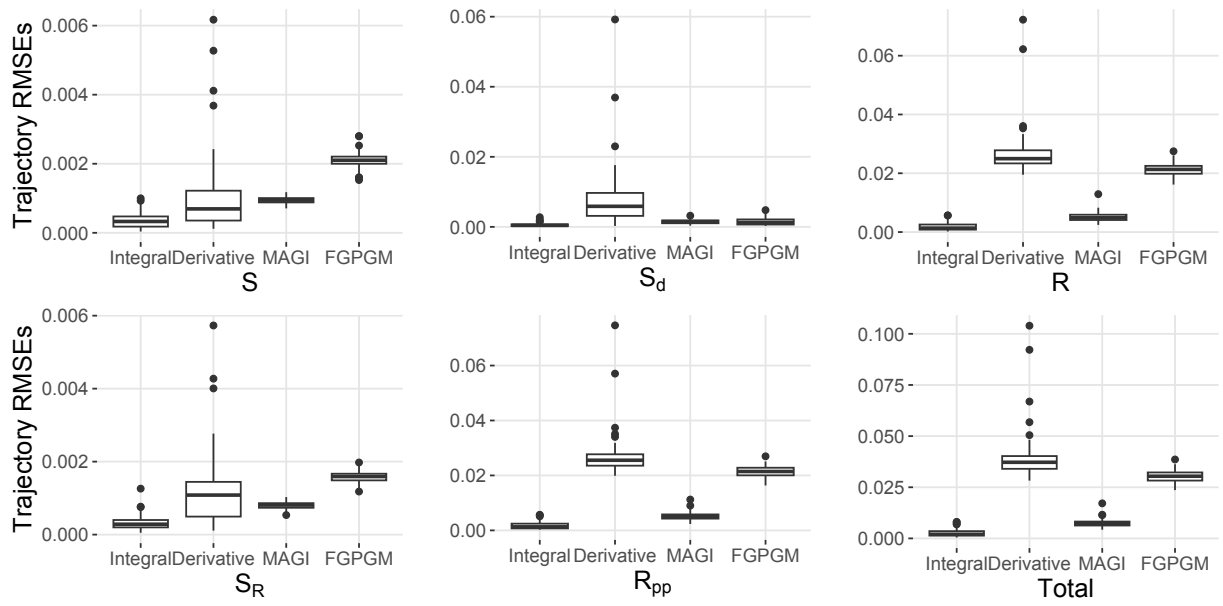


Figure B.10: Estimated trajectory RMSEs for each component in the protein transduction system, over 100 simulation runs with low noise.

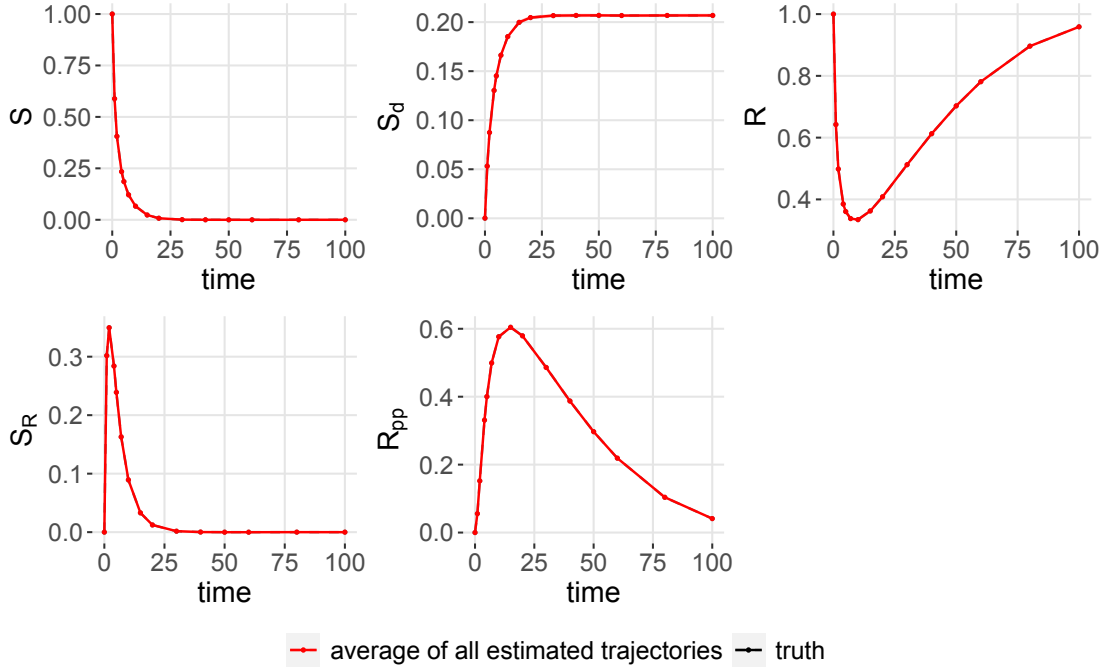


Figure B.11: Estimated trajectories for each component in the protein transduction system, over 100 simulation runs with low-noise setting. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

Under the high noise setting, our method still outperforms all other methods in terms of trajectory RMSEs for all components, as indicated in Table B.3 and Figure B.12. Although the derivative method has lower trajectory RMSEs for components S and S_R than MAGI, it has higher trajectory RMSEs for components S_d and R than FGPGM. Figure B.13 displays the average estimated ODE solution across 100 simulation runs along with the 95% pointwise intervals. With slightly wider 95% pointwise intervals than the low noise setting, especially for component S_d , our method still recovers the system well with reasonable inferences for the ODE solution.

In addition, while the FN system features the periodic oscillations, the protein transduction system features some rapid/sharp changes, as shown in Figures B.11 and B.13. Consequently, derivative estimation might be less efficient for handling such sharp changes, and our integral-based method outperforms derivative-based ones more noticeably in this protein transduction system than in the FN system.

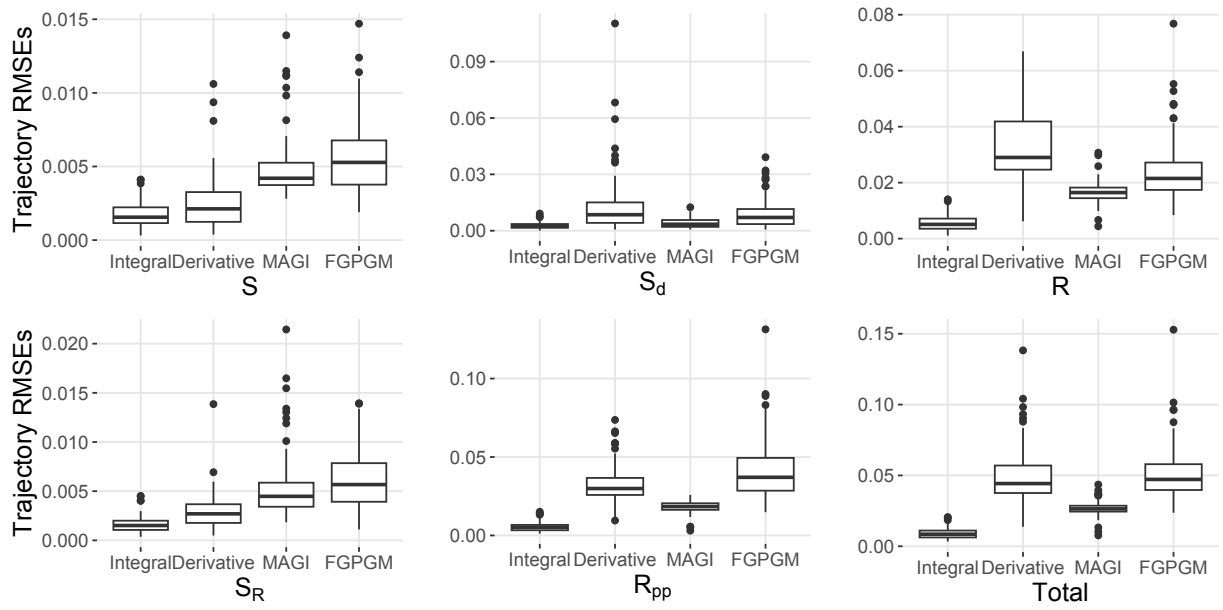


Figure B.12: Estimated trajectory RMSEs for each component in the protein transduction system, over 100 simulation runs with high noise.

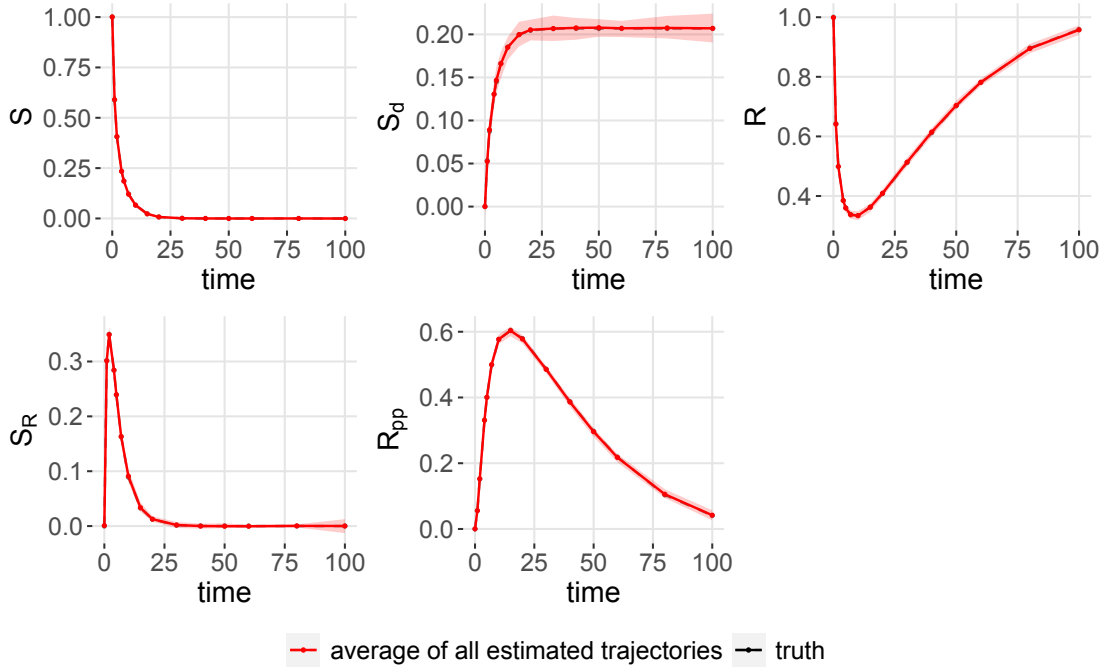


Figure B.13: Estimated trajectories for each component in the protein transduction system, over 100 simulation runs with high-noise setting. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

Similar to the FN system, Table B.4 reports the average estimated norm for each component under both noise settings. With a higher noise level, the differences between the estimated and true norm become more obvious for each method. Moreover, the estimated norms of our method are generally closer to the true norms than those of three other competitors, suggesting that our method can outperform other methods when reconstructing the trajectories.

Noise level	Method	S	S_d	R	S_R	R_{pp}
	True	3.470	6.328	22.420	2.411	11.115
Low-noise, $\sigma = 0.001$	Integral	3.471	6.324	22.406	2.410	11.108
	Derivative	3.481	6.505	21.925	2.434	11.077
	MAGI	3.488	6.283	22.327	2.413	11.126
	FGPGM	3.525	6.351	21.980	2.394	11.145
High-noise, $\sigma = 0.01$	Integral	3.483	6.342	22.403	2.418	11.089
	Derivative	3.493	6.601	21.867	2.452	11.096
	MAGI	3.558	6.254	22.391	2.479	11.171
	FGPGM	3.602	6.373	22.070	2.513	11.816

Table B.4: Average norm of each estimated component of the protein transduction system across the 100 simulation runs.

Table B.5 displays the average running time of each method for the protein transduction system under both noise settings. Similar to the FN system, the derivative method is expected to be faster than our integral method. Our method is about 5 times faster than MAGI and about 106 times faster than FGPGM under both noise settings when they are implemented on the same processor. In summary, our method is considerably more efficient than the competitors.

Noise	Integral	Derivative	MAGI	FGPGM
Low	1.42	0.23	7.89	175.80
High	1.65	0.22	7.50	175.03

Table B.5: Average running time (minutes) of each method for the protein transduction system across 100 simulation runs.

B.1.3 Order of basis functions

To investigate the impact of the order of the basis functions on the estimation process, the proposed method is applied to the FN system with 41 observations with different B-spline orders. As described in Section 2.3.1 of the main text, we originally place 83 cubic B-spline basis functions within the time interval for this simulation scenario. As a comparison, we now place 84 and 85 B-spline basis functions of order five and six, respectively, while the other settings are not changed. Similarly, 100 independent simulation trials are run to assess the variability of the parameter estimates.

Table B.6 presents parameter estimation results for the proposed method with B-splines of different orders, and Figure B.14 displays the boxplot for parameter RMSEs. In spite of the higher order of basis functions, there are only subtle differences in the parameter estimates and RMSEs.

Order	a	b	c
4	0.20(0.02)	0.26(0.10)	2.95(0.07)
5	0.20(0.02)	0.25(0.10)	2.94(0.07)
6	0.20(0.02)	0.27(0.11)	2.95(0.07)

Table B.6: Mean of the parameter estimates with RMSE in parentheses for the FN system with 41 observations using the proposed method across the 100 simulation runs.

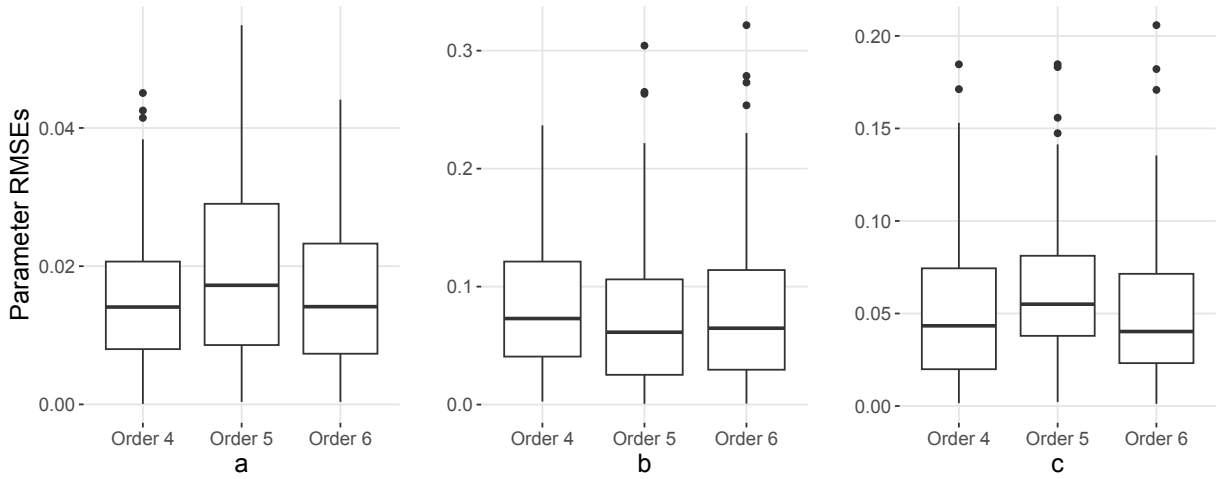


Figure B.14: Estimated parameter RMSEs in the FN system using the proposed method, over 100 simulation runs with 41 observations.

Table B.7 shows the median and IQR of trajectory RMSEs of the proposed method with different order B-splines, and Figure B.15 presents the corresponding boxplots. Though there exists a larger discrepancy in the trajectory RMSEs than the parameter RMSEs between different orders of basis functions, the performance of estimated components is still quite similar for different orders.

Order	V	R	Total
4	0.086(0.055)	0.045(0.040)	0.104(0.057)
5	0.103(0.050)	0.049(0.032)	0.117(0.050)
6	0.083(0.044)	0.043(0.034)	0.095(0.052)

Table B.7: Median of trajectory RMSEs (with IQR in parentheses) for each component of the FN system with 41 observations using the proposed method across 100 simulation runs.

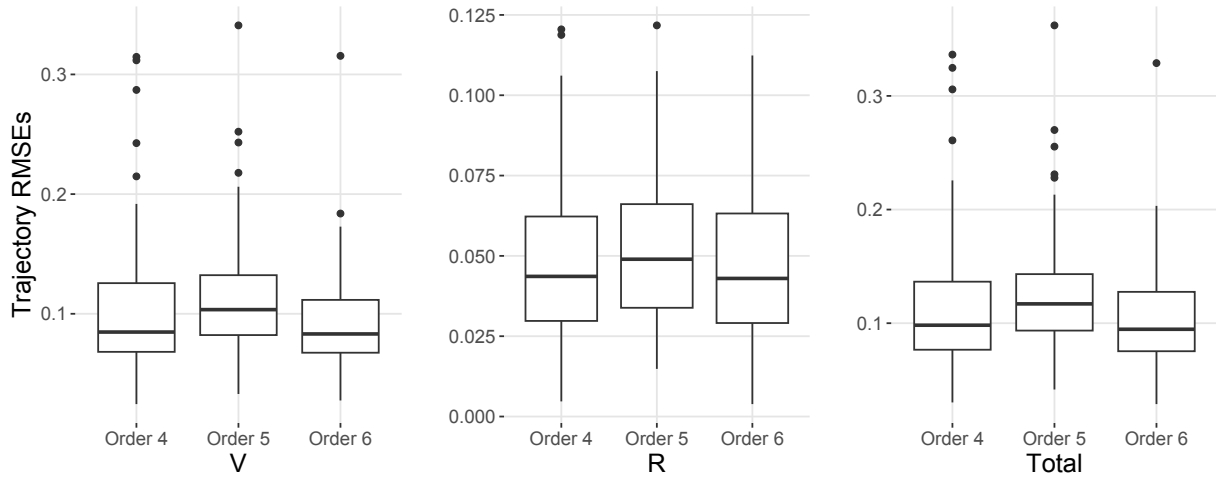


Figure B.15: Estimated trajectory RMSEs for each component in the FN system using the proposed method, over 100 simulation runs with 41 observations.

Furthermore, we calculate the average norm for each component and summarize the results in Table B.8, which indicates different orders of basis functions yield similar average norms. Therefore, we conclude that the order of the basis functions does not have a remarkable impact on estimation results. However, based on the average running time displayed in Table B.9, we recommend using cubic B-splines, i.e., the B-spline of order four.

Order	V	R
True	47.079	21.972
4	46.631	21.848
5	46.731	21.955
6	46.635	21.862

Table B.8: Average norm for each estimated component of the FN system with 41 observations using the proposed method across 100 simulation runs.

Order	4	5	6
Run time	3.28	4.98	4.78

Table B.9: Average running time (minutes) of the FN system with 41 observations using the proposed method across 100 simulation runs.

B.1.4 Consistency of selecting λ

To demonstrate the selection consistency of λ in our smoothing parameter selection algorithm (Section 2.2.3), we first plot the frequency of selected λ 's in the FN system across 100 simulation runs with 41 observations. As shown in Figure B.16, the selection algorithm tends to frequently select $\lambda = 10^3$ and $\lambda = 10^4$.

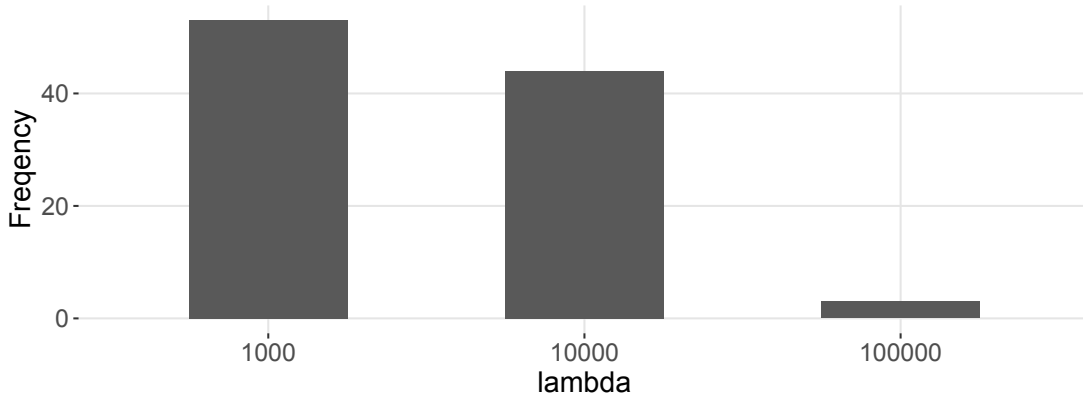


Figure B.16: Selected λ in the FN system using the proposed method, over 100 simulation runs with 41 observations.

To showcase the asymptotic behavior of the function estimate, we pick the simulation runs that select $\lambda = 10^3$ and increase these λ 's to slightly larger ones, i.e., $\lambda = 5 \times 10^3$ and $\lambda = 10^4$ and compare the performance. Instead of calculating the trajectory RMSEs as in Section 2.3.2, we calculate

$$\text{trajectory RMSE}_\lambda = \sqrt{\int_{t_1}^{t_J} [\tilde{x}(t) - \tilde{x}(t)_{\lambda=10^3}]^2 dt}$$

where $\tilde{x}(t)$ and $\tilde{x}(t)_{\lambda=10^3}$ are the reconstructed trajectories of the larger λ and $\lambda = 10^3$, respectively. Table B.10 reports the result of these new trajectory RMSEs for each component, where the differences among these λ 's are relatively small, especially for component R . Moreover, we calculate the average norm for each component obtained from these λ 's and present the result in Table B.11. Though we increase the value of λ , our method produces similar average norms.

λ	V	R	Total
5×10^3	0.074(0.058)	0.022(0.015)	0.079(0.063)
10^4	0.078(0.075)	0.026(0.014)	0.080(0.076)

Table B.10: Median of trajectory RMSEs between $\lambda = 10^3$ and $\lambda = 5 \times 10^3/10^4$ (with IQR in parentheses) for each component of the FN system with 41 observations across the 100 simulated runs.

λ	V	R
True	47.079	21.972
10^3	46.484	21.831
5×10^3	46.516	21.811
10^4	46.372	21.741

Table B.11: Average norm for each estimated component of the FN system with 41 observations using the proposed method across 100 simulation runs.

Figures B.17, B.18 and B.19 display the estimated trajectories with 95% pointwise credible intervals for each component obtained from $\lambda = 10^3$, 5×10^3 and 10^4 , respectively. As λ increases, the central 95% pointwise credible intervals are expected to be slightly narrower. Despite this, the estimated trajectories of different λ recover the system fairly

well and do not differ significantly. Consequently, all of these results demonstrate the consistency of selecting λ and the asymptotic behavior of the function estimate.

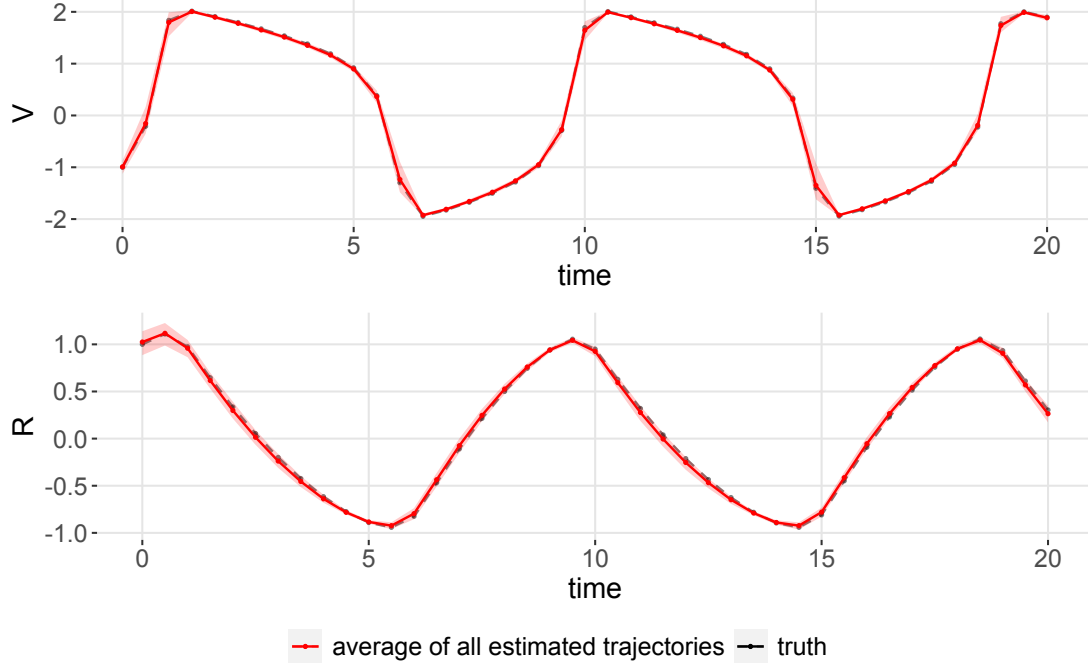


Figure B.17: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations with $\lambda = 10^3$. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

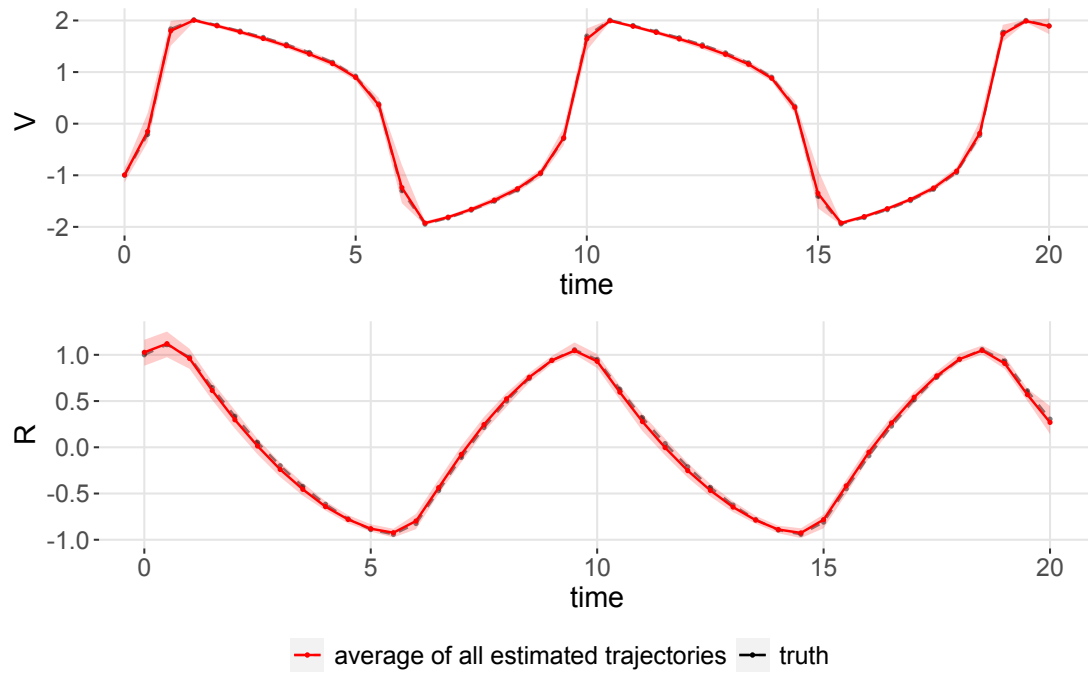


Figure B.18: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations with $\lambda = 5 \times 10^3$. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

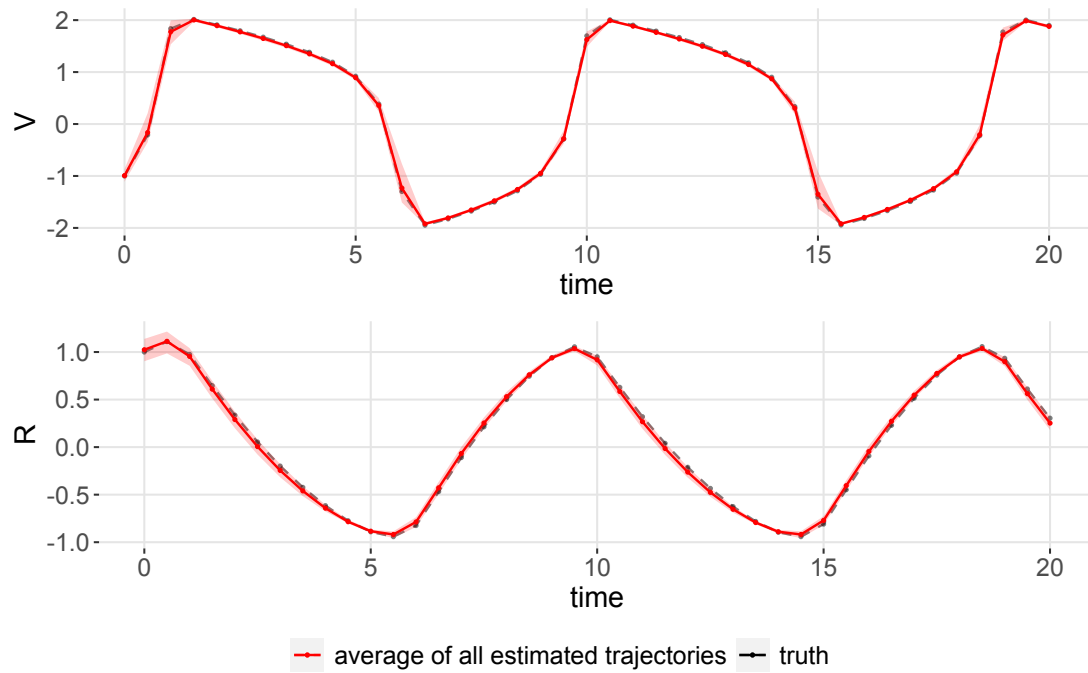


Figure B.19: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system for 41 observations with $\lambda = 10^4$. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

B.1.5 Irregular Observations

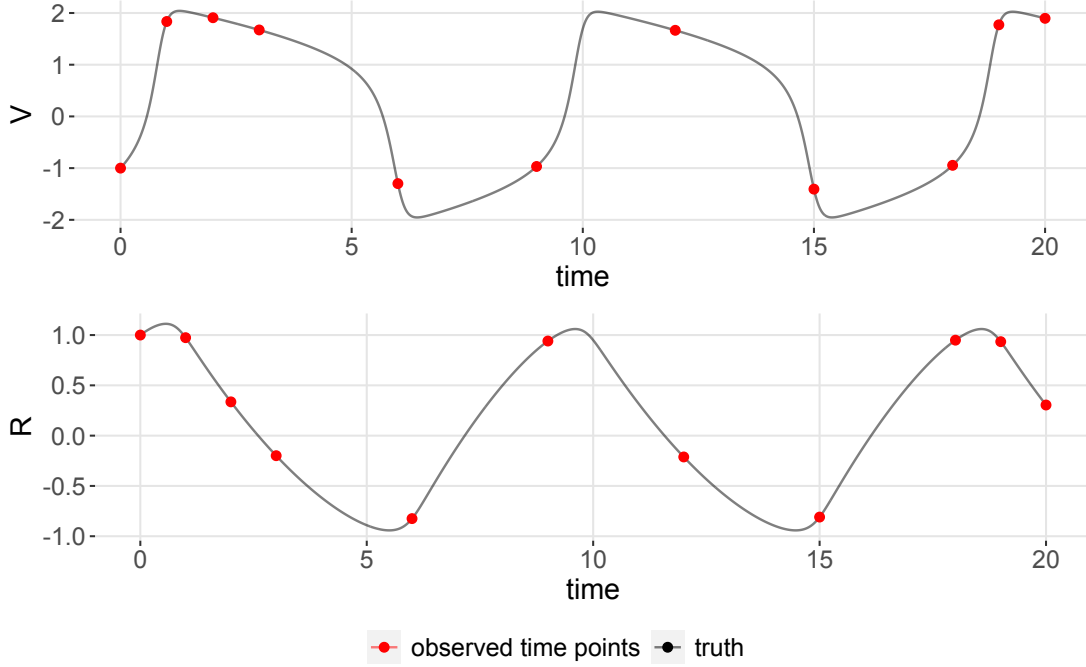


Figure B.20: True trajectories for V (top panel) and R (bottom panel) in the FN system and the spacing of the 11 irregular observations. The black line represents the true trajectories and the red points are the locations of the observed time points.

To showcase the applicability of our method to sparse and irregularly observed data, we generate 11 unequally-spaced noisy observations from the FN system, where $t = \{0, 1, 2, 3, 6, 9, 12, 15, 18, 19, 20\}$, as illustrated in Figure B.20. Visually, we notice that the time spacing of 3 for $t = 3, 6, 9, 12, 15, 18$ is very sparse and the observations will fail to capture some of the key oscillation behaviour. For our method, we place one knot at each time point and insert one additional knot between successive observation time points, which results in 23 basis functions with 21 knots. All other settings, including the setup of the simulation and implementation of other competitors, are identical to the ones for 11 equally-spaced observations in Section 2.3 of the main text.

Table B.12 summarizes the results of parameter estimates, while Figure B.21 displays the boxplot of parameter RMSEs. Similar to the case of equally-spaced observations, all methods suffer from the poor parameter estimation. Particularly, our method has the

lowest parameter RMSE for a . Although FGPGM has the lowest parameter RMSEs for b and c among all methods, its parameter RMSEs vary widely.

Method	a	b	c
Integral	0.11(0.10)	0.67(0.47)	1.03(1.97)
Derivative	0.07(0.13)	0.64(0.44)	0.96(2.04)
MAGI	0.13(0.11)	0.65(0.46)	1.16(1.88)
FGPGM	0.50(0.34)	0.42(0.32)	1.81(1.43)

Table B.12: Mean of the parameter estimates with RMSE in parentheses for the FN system with 11 irregular observations across the 100 simulation runs.

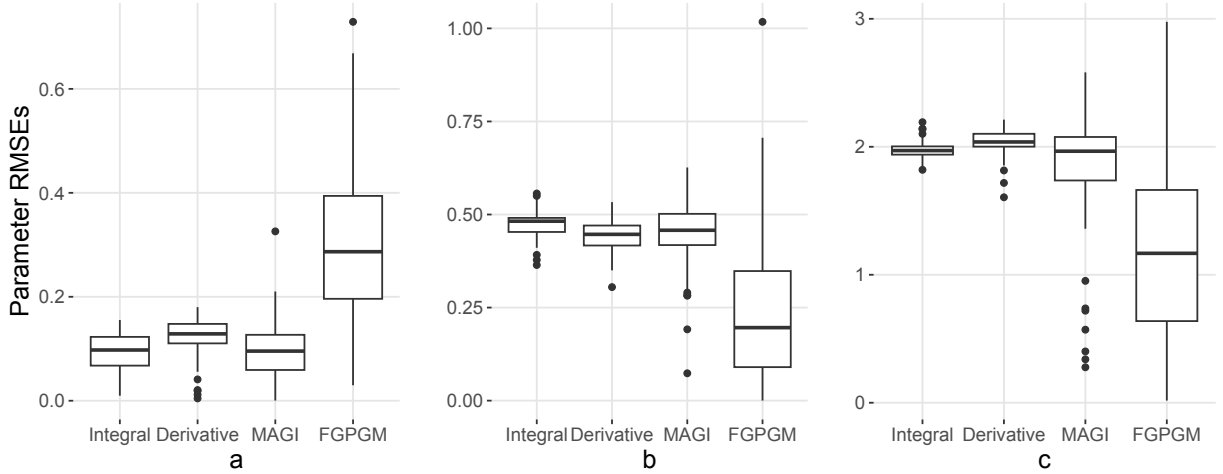


Figure B.21: Estimated parameter RMSEs for a, b and c respectively in the FN system, over 100 simulation runs with 11 irregular observations.

Table B.13 shows the median and IQR of trajectory RMSEs for each component, while Figure B.22 presents the boxplot of trajectory RMSEs. The pattern is similar to the case of equally-spaced observations: our method is superior to the other three competitors and has a marginally lower total trajectory RMSE. Despite having low parameter RMSEs, FGPGM has significantly higher trajectory RMSEs than other methods. Table B.14 presents the average norms of each estimated component, while the pattern is similar to that of the equally-spaced case, except that FGPGM has the closest norms of each estimated component to the true ones compared with the other methods. The reason could be that FGPGM

has the largest range of estimated norms, so the average norms can be close to the true ones.

Method	V	R	Total
Integral	0.983(0.151)	0.220(0.084)	1.021(0.143)
Derivative	1.026(0.196)	0.202(0.111)	1.052(0.166)
MAGI	0.991(0.317)	0.237(0.280)	1.050(0.246)
FGPGM	1.444(0.532)	0.833(0.453)	1.671(0.638)

Table B.13: Median of trajectory RMSEs (with IQR in parentheses) for each component of the FN system with 11 irregular observations across 100 simulation runs.

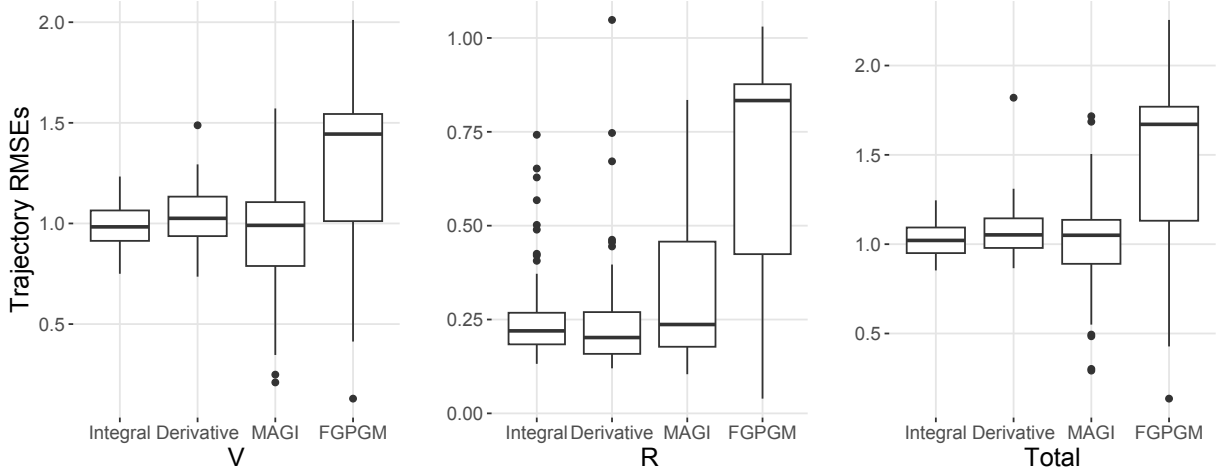


Figure B.22: Estimated trajectory RMSEs for V (left panel), R (middle panel) and total (right panel) in the FN system, over 100 simulation runs with 11 irregular observations.

Method	V	R
True	47.079	21.972
Integral	24.450	24.640
Derivative	21.874	23.531
MAGI	25.785	23.191
FGPGM	36.754	22.383

Table B.14: Average of the norms of each estimated component in the FN system with 11 irregular observations across 100 simulation runs.

Figure B.23 depicts the estimated trajectories of our method for each component as well as the 95% pointwise intervals. The result is still similar to the equally-spaced case: in contrast to component R , the 95% pointwise intervals of component V do not fully cover the true trajectory. Specifically, the estimated trajectory of V does not provide an appropriate recovery of the system, which further explains the difficulty of estimating a nonlinear system.

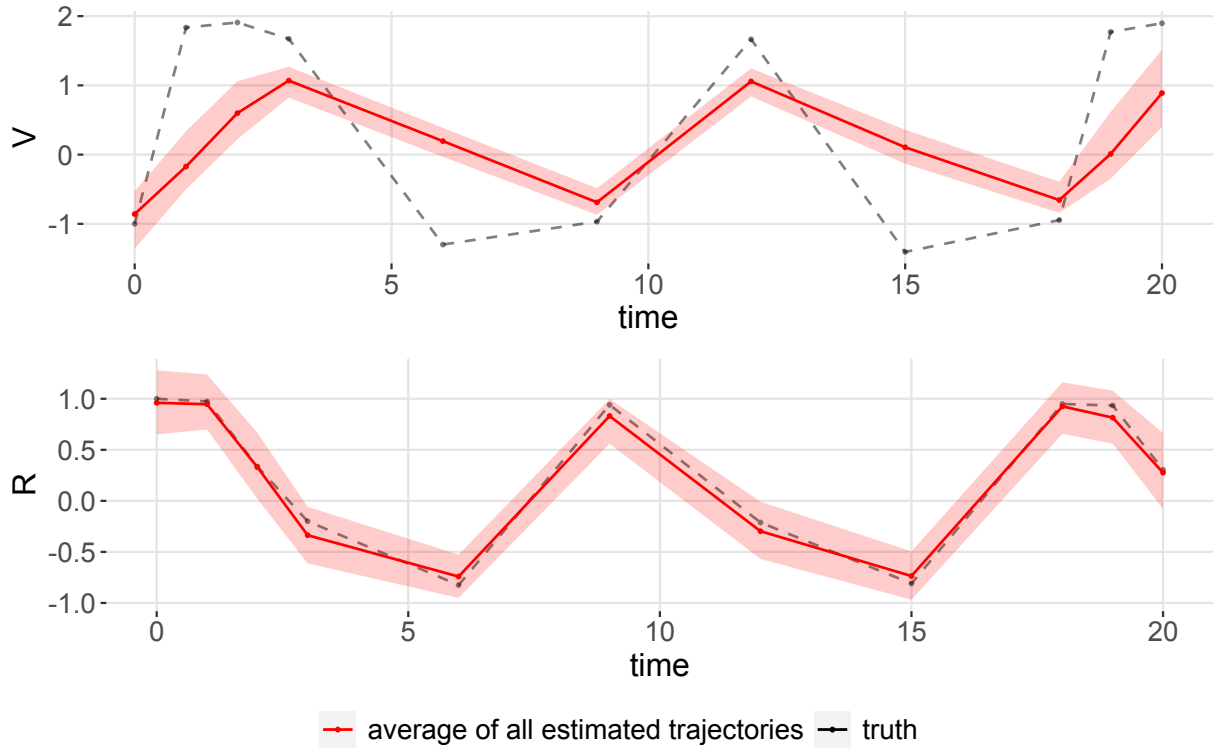


Figure B.23: Estimated trajectories for V (top panel) and R (bottom panel) in the FN system, over 100 simulation runs with 11 irregular observations. The black dashed line and the red solid line represent the true and the average estimated trajectories, respectively. The red shaded area represents the pointwise central 95% intervals of all estimated trajectories.

B.2 Real data analysis: additional results

As mentioned in Section 2.4 of the main text, we apply the Lotka-Volterra (LV) model (Lotka, 1932) to fit the data for the number of Snowshoe hares and Canadian lynxes traded in the years 1908-1928 (Odum, 1953). As described in Section 2.2.4, we use NUTS as the sampling algorithm for our proposed method. To ensure convergence as well as a low computing cost, we set the maximum tree depth to 13 and the target average proposal acceptance probability to 0.85.

We use the same comparison methods as in the simulation study. Particularly, for MAGI, we follow the recommendation of authors and use their R package *magi*: 81 discretizations are used to approximate the random variable that quantifies the difference between GP derivatives and ODE systems, and 20,000 HMC iterations are run, including the first half of the iterations as a warm-up. Additionally, each HMC iteration uses 100 leapfrog steps. For FGPGM, we run their software: the standard deviation γ that adjusts the potential model mismatch between the ODE output and GP derivatives is set to 0.3 and a squared exponential kernel is used. 300,000 MCMC iterations are run, including the first half of the iterations as a warmup. The Bayesian collocation with derivative estimation is conducted with the exact same setting as our method described in Section 2.2.4.

Table B.15 summarizes the parameter estimation results for these three competitors. The MAGI, derivative-based, and integral-based methods do not produce significantly different parameter estimates; FGPGM, however, differs greatly in terms of the parameter estimates.

Estimate	Derivative	MAGI	FGPGM
θ_1	0.683 (0.427, 0.961)	0.735 (0.503, 0.981)	0.642 (0.309, 0.991)
θ_2	0.027 (0.018, 0.037)	0.030 (0.023, 0.040)	0.025 (0.013, 0.037)
θ_3	0.527 (0.368, 0.764)	0.484 (0.345, 0.718)	0.245 (0.050, 0.451)
θ_4	0.014 (0.009, 0.021)	0.013 (0.009, 0.019)	0.008 (0.003, 0.013)
x_{01}	25.311 (19.446, 34.272)	24.995 (20.721, 30.204)	20.094 (13.716, 26.537)
x_{02}	11.846 (6.732, 15.965)	12.239 (7.363, 15.887)	4.001 (0.056, 7.947)

Table B.15: Parameter estimates with 95% credible interval in parentheses, when using the LV model to fit the hare and lynx data for 1908-1928. Here x_{01} and x_{02} denote the estimated initial conditions for x_1 and x_2 , respectively.

In order to demonstrate the performance of three competitors in estimating ODE solutions, we consider the estimated $x_i(t)$ based on the spline approximation for collocation

methods, while for GP-based methods, we consider the sampled GP trajectory. Figures B.24 and B.25 depict the estimated trajectories from the derivative method and MAGI, respectively. Both methods provide reasonable fits for the time-varying numbers of lynxes and hares, with appropriate 95% pointwise credible intervals. In contrast, the estimated trajectories from FGPGM completely fit the noisy observations rather than the fitted LV system, and the 95% pointwise credible intervals almost encompass all the observations. As shown in Table B.16, MAGI and the derivative-based method are slightly faster than our proposed integral-based method in terms of computations, while FGPGM takes over an hour to run.

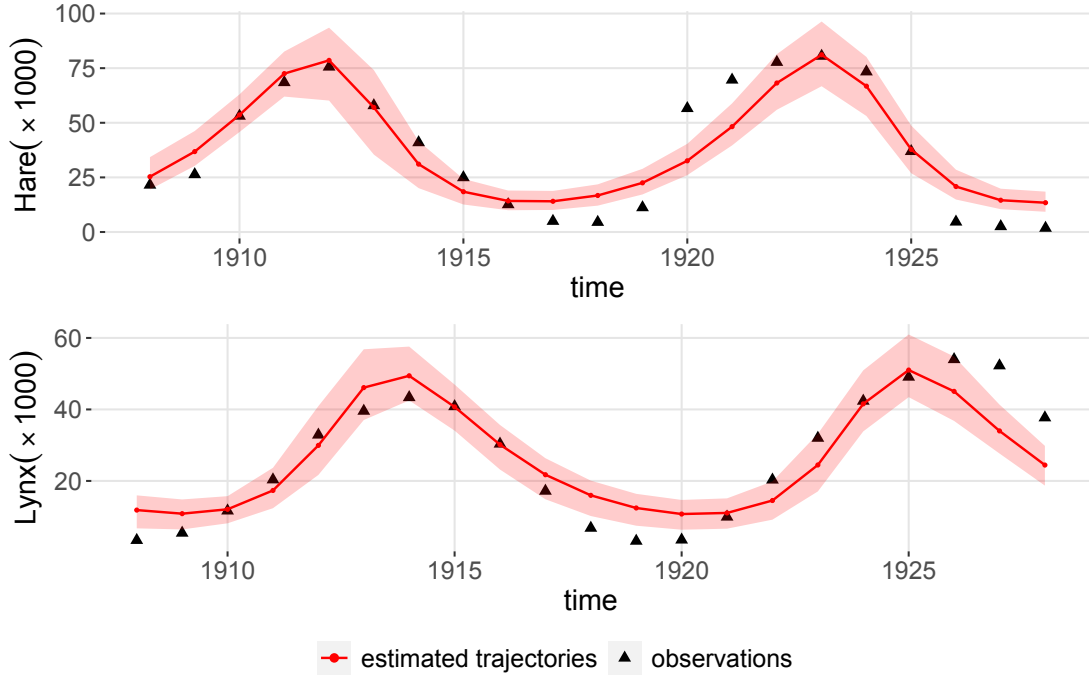


Figure B.24: Estimated trajectories from the derivative method for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

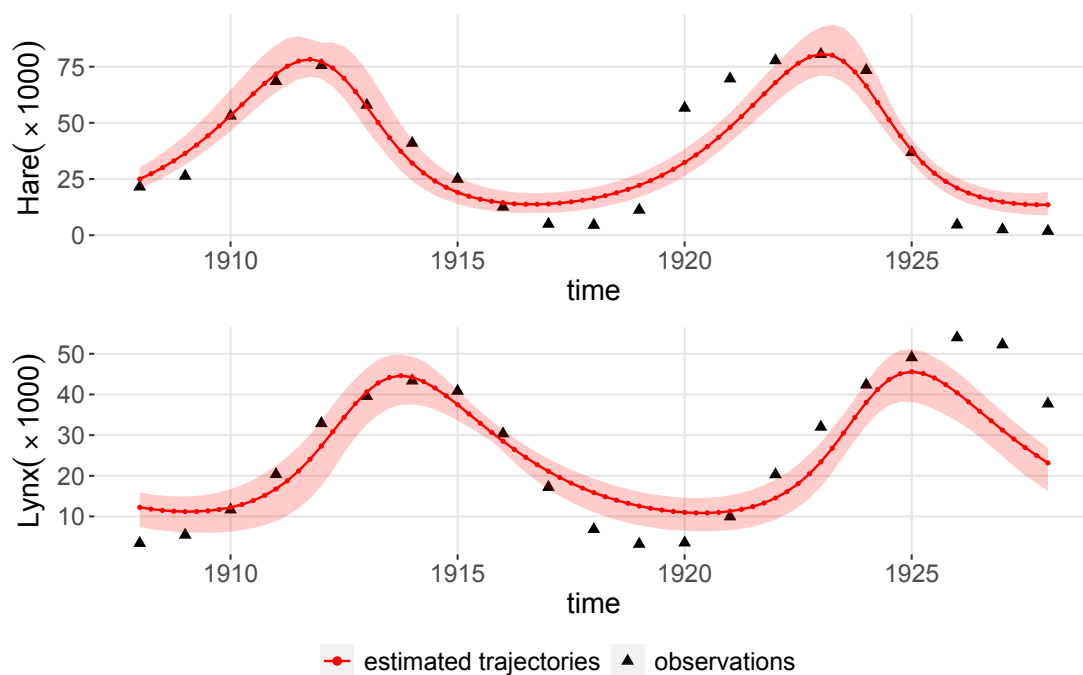


Figure B.25: Estimated trajectories from MAGI for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

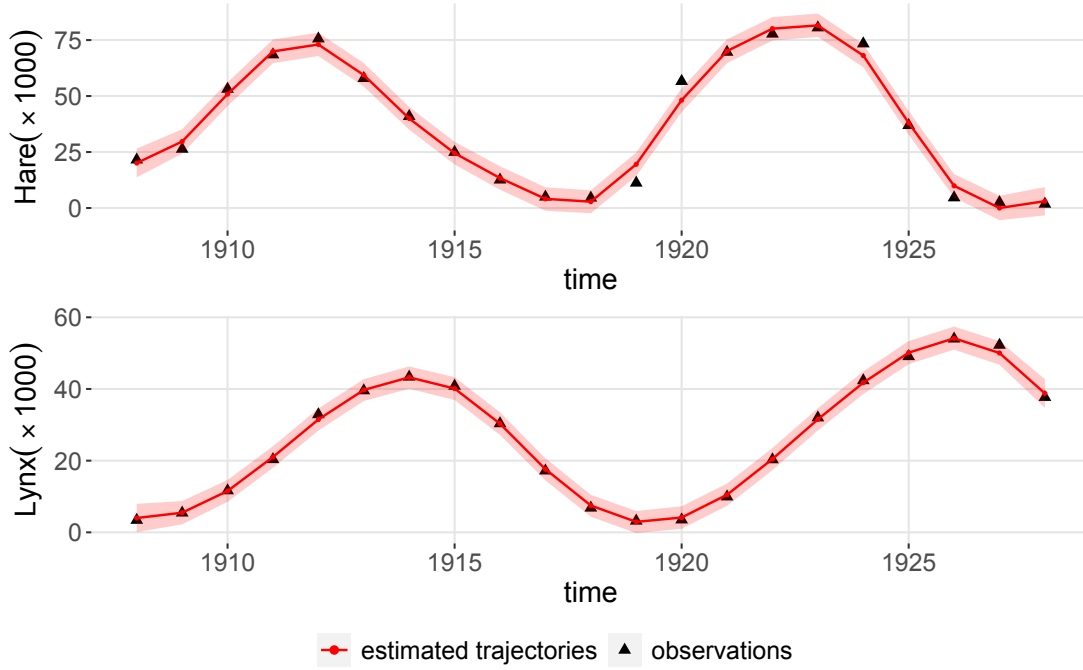


Figure B.26: Estimated trajectories from FGPGM for the numbers of Canadian lynxes (bottom) and snowshoe hares (top) in the years 1908-1928. The black triangles represent the real data and the red solid line is the estimated trajectory. The red shaded area is a central 95% pointwise credible interval for the estimated trajectory.

	Integral	Derivative	MAGI	FGPGM
Run time	5.76	1.23	3.35	87.00

Table B.16: Running time (minutes) of each method when using the LV model to fit the hare and lynx data for 1908-1928.

To further assess the performance of the different methods, for each method we calculate the RMSEs between the reconstructed trajectories and the actual observations, as displayed in Table B.17. Similar to the reconstructed trajectories introduced in Section 2.3.2 of the main text, we numerically solve the trajectories with the estimated parameters and initial conditions using the LV model. Our proposed integral method has the lowest RMSEs among all the methods, and thus has the advantage of providing the best fit to the real data.

Method	Integral	Derivative	MAGI	FGPGM
Lynx	10.563	10.670	10.599	45.998
Hare	7.834	8.177	8.439	29.103

Table B.17: RMSEs between the reconstructed trajectories and the observations using the LV model and the hare and lynx data for 1908-1928.

Appendix C

Additional Results for Chapter 3

C.1 Choice of quadrature points

To yield an exact result for integrating a function that can be well-approximated by a polynomial of degree $(2K - 1)$, the Gaussian quadrature rule requires at least K quadrature points (Golub and Welsch, 1969). With $(M + 1)$ quadrature points, the Gaussian quadrature is exact for integrating products of B-splines with degree M between two adjacent knots (de Boor et al., 1976). Hence, the Gaussian quadrature needs $(M + 1)(L - 2)$ quadrature points to be exact for the outer integral (15), where $(L - 2)$ is the number of interior knots. However, Xu et al. (2024) found that using roughly $(M + 1)(L - 2)/2$ quadrature points can yield results close to the exact integration. While increasing the number of quadrature points improves approximation accuracy, it also raises computational cost. Therefore, we decide to use approximately $(M + 1)(L - 2)/2$ quadrature points for the outer integral in both the simulation studies and real data analysis.

C.2 Simulation studies: additional results

C.2.1 Additive ODEs

As mentioned in Section 3.3.3 of the main text, the pointwise coverage probabilities of our proposed method across 100 simulation runs for $\text{SNR} = 25$ and 4 are shown in Figures C.1 and C.2, respectively. Similar to Table 3.1 in the main text, the results for 21 observations with different noise settings are displayed in Table C.1.

Similar to the case with 41 observations, we present the pointwise coverage probabilities for each trajectory in the 21 observations case. The pointwise coverage probabilities of our proposed method in the simulated ODE system across 100 simulation runs for SNR=25, 10 and 4 are depicted in Figure C.3, C.4, and C.5, respectively. As the observations become sparser, the trajectory estimation error increases. Hence, the pointwise coverage probabilities will be lower than those in the setting with 41 observations, especially for the nonlinear trajectories $\mathbf{x}_1$ to $\mathbf{x}_6$. Nevertheless, these figures show a similar pattern to those in the 41 observations case as the noise level increases.

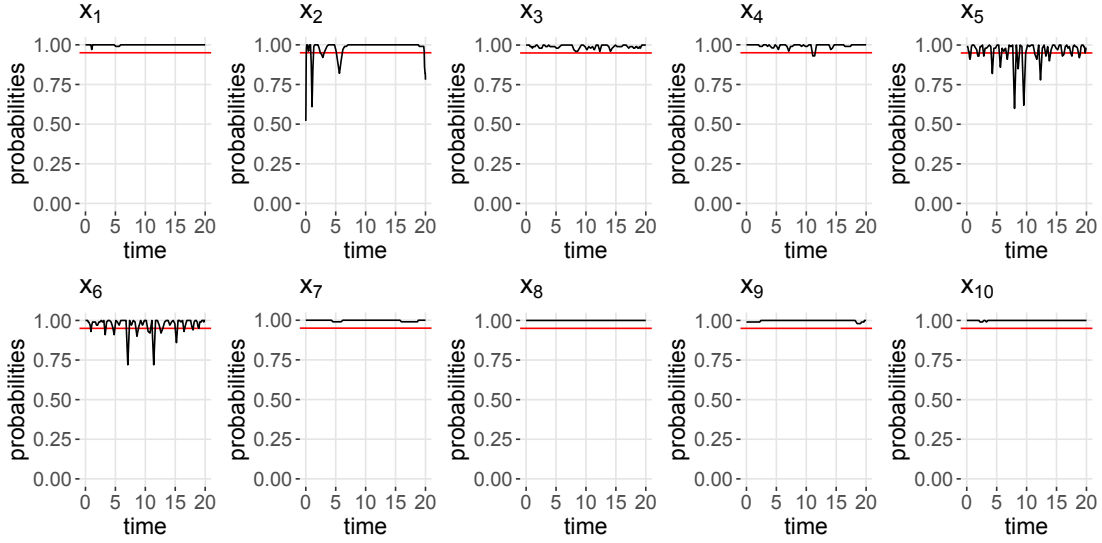


Figure C.1: The pointwise coverage probabilities of each trajectory in the simulated ODE system for 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.

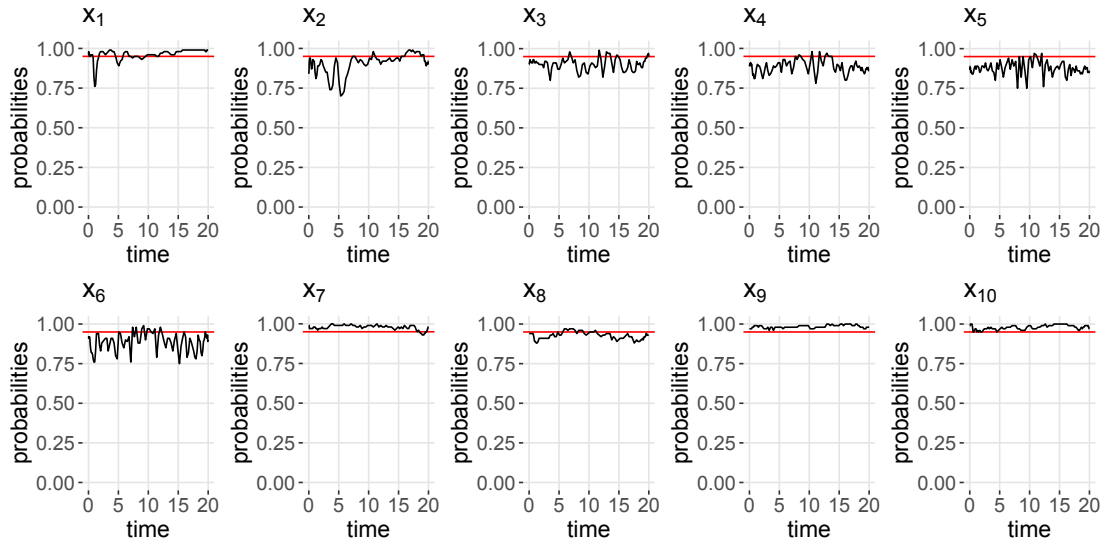


Figure C.2: The pointwise coverage probabilities of each trajectory in the simulated ODE system for 41 observations over 100 simulation runs when $\text{SNR}=4$ (black line). The red horizontal line indicates the nominal level 95%.

Table C.1: Comparison of methods for 21 observations in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.

SNR	Method	$\text{MSE}(\hat{\mathbf{x}})$ ($\times 10^{-2}$)	$\text{MSE}_{\mathcal{A}}(\hat{f})$	$\text{MSE}_{\mathcal{A}^c}(\hat{f})$ ($\times 10^{-2}$)	$\text{MSE}(\hat{f})$	TPR	FPR
25	Integral	0.859	0.186	0.018	0.015	0.948	0.164
		(0.010)	(0.027)	(0.059)	(0.002)	(0.071)	(0.043)
	JADE	0.569	0.193	0.218	0.018	0.874	0.039
		(0.048)	(0.050)	(0.314)	(0.007)	(0.045)	(0.026)
	GRADE	0.849	0.384	3.200	0.060	0.999	0.624
		(0.053)	(0.098)	(2.053)	(0.023)	(0.013)	(0.112)
10	Integral	1.391	0.229	0.093	0.019	0.881	0.127
		(0.017)	(0.047)	(0.138)	(0.005)	(0.086)	(0.035)
	JADE	1.330	0.274	0.274	0.031	0.814	0.058
		(0.229)	(0.111)	(2.396)	(0.028)	(0.113)	(0.043)
	GRADE	2.015	0.559	8.654	0.124	0.999	0.815
		(0.230)	(0.149)	(6.968)	(0.070)	(0.013)	(0.066)
4	Integral	4.679	0.337	0.373	0.030	0.695	0.074
		(0.844)	(0.074)	(0.526)	(0.010)	(0.122)	(0.031)
	JADE	6.753	0.550	3.352	0.075	0.663	0.114
		(3.549)	(0.333)	(6.273)	(0.067)	(0.185)	(0.076)
	GRADE	7.473	1.470	17.890	0.282	0.999	0.852
		(0.169)	(0.442)	(4.837)	(0.080)	(0.013)	(0.063)

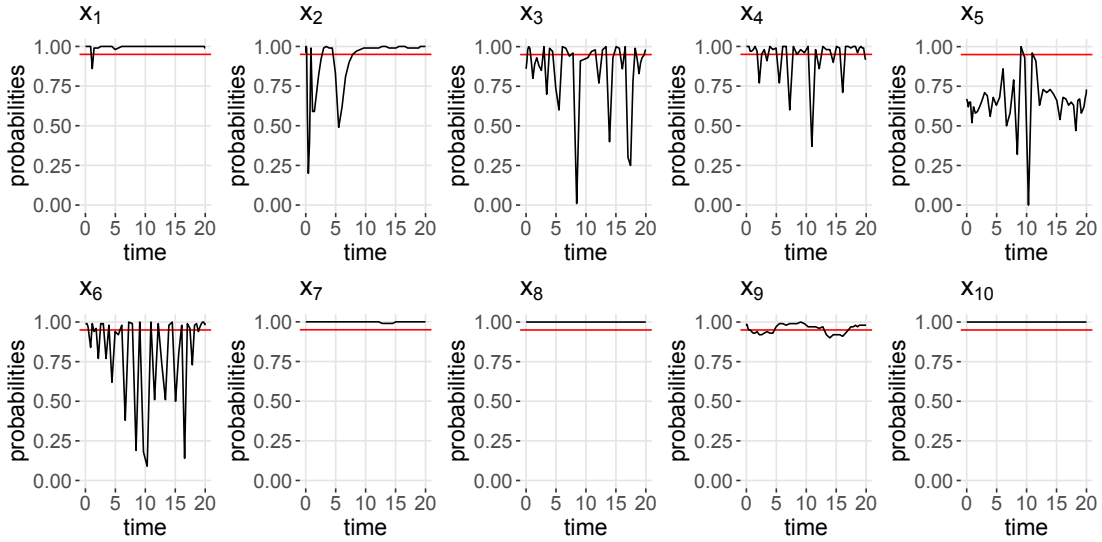


Figure C.3: The pointwise coverage probabilities of each trajectory in the simulated ODE system for 21 observations over 100 simulation runs when $\text{SNR}=25$ (black line). The red horizontal line indicates the nominal level 95%.

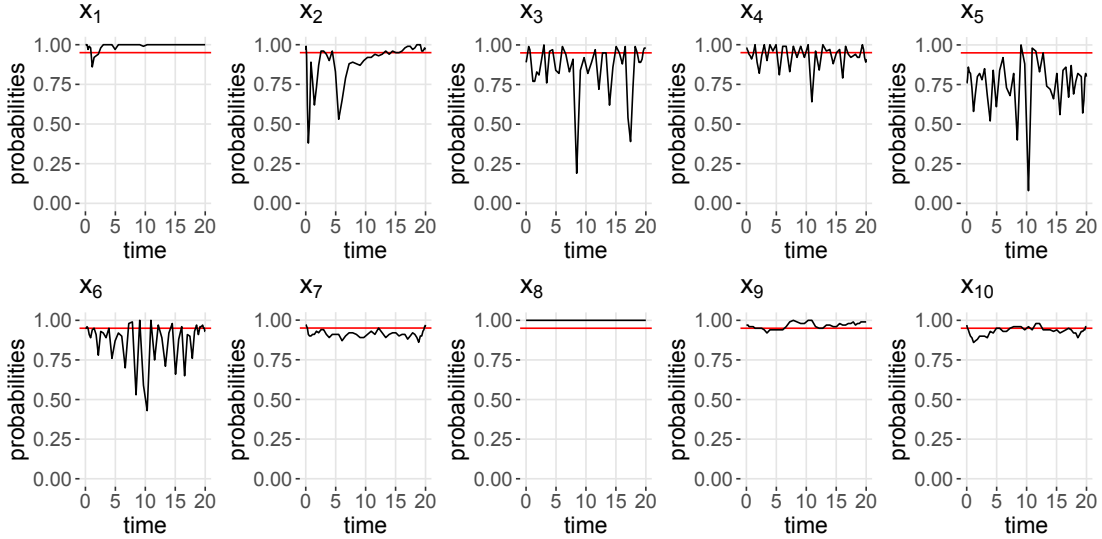


Figure C.4: The pointwise coverage probabilities of each trajectory in the simulated ODE system for 21 observations over 100 simulation runs when $\text{SNR}=10$ (black line). The red horizontal line indicates the nominal level 95%.

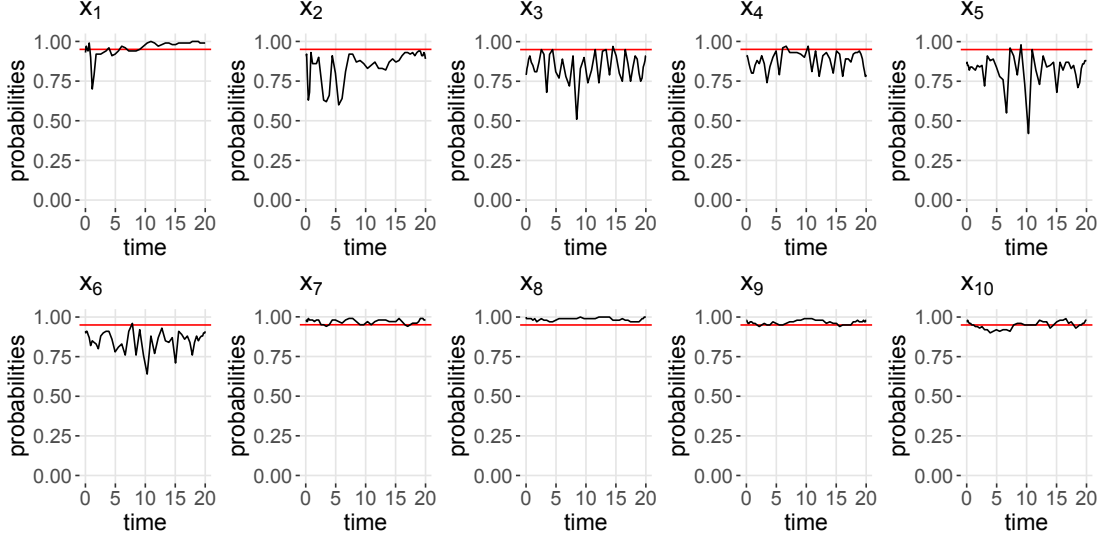


Figure C.5: The pointwise coverage probabilities of each trajectory in the simulated ODE system for 21 observations over 100 simulation runs when $\text{SNR}=4$ (black line). The red horizontal line indicates the nominal level 95%.

Regarding computation costs under different sampling frequencies, the average running time of the proposed method is approximately 87.82 and 45.67 minutes under the settings of 41 and 21 observations, respectively. For 41 observations, twice as many quadrature points are used as for 21 observations, so we can expect a nearly doubled running time. To investigate the computation time of larger systems, our method is applied to the simulation system (3.15) in Section 3.3.1 of the main text with $k = 1, \dots, 8$ and $\text{SNR} = 25$. For $k = 1, \dots, 5$, we set the initial values and parameters as in the simulation settings in Section 3.3.1, while the initial value and parameters of $k = 6, \dots, 8$ are the same as those of $k = 4$ or 5. In other words, the newly added variables are all linear in time t . Consequently, this new ODE system can still be expressed as additive ODEs (3), with only 8 out of 256 functions $f_{ip}, i, p = 1, \dots, 16$ being non-zero. Similarly, we run 100 independent simulation trials for 41 observations case with the same setup as in Section 3.3.1. The average running time of our method is approximately 168.81 minutes, roughly twice the time for the original simulation system.

Table C.2: Comparison of number of equations for 41 observations with SNR = 25 in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.

k	$\text{MSE}(\hat{\mathbf{x}})$ ($\times 10^{-2}$)	$\text{MSE}_{\mathcal{A}}(\hat{f})$	$\text{MSE}_{\mathcal{A}^c}(\hat{f})$ ($\times 10^{-2}$)	$\text{MSE}(\hat{f})$	TPR	FPR
5	0.190 (0.020)	0.081 (0.006)	0.003 (0.008)	0.007 (0.001)	0.999 (0.013)	0.167 (0.043)
8	0.146 (0.002)	0.081 (0.008)	0.002 (0.004)	0.002 (0.001)	0.983 (0.056)	0.227 (0.034)

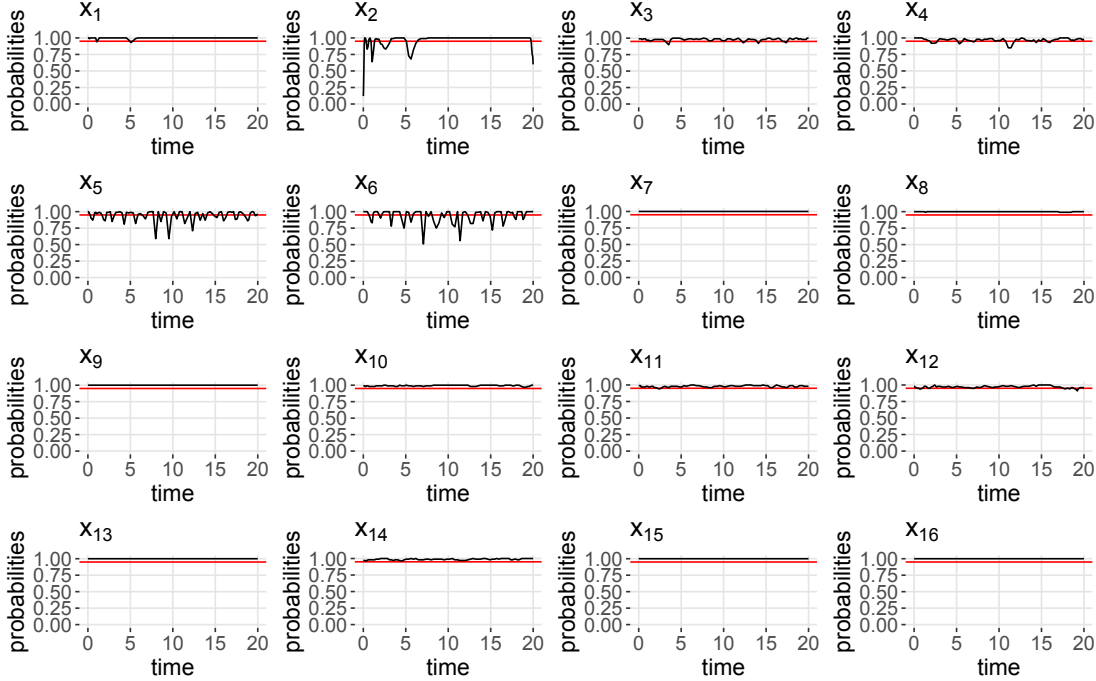


Figure C.6: The pointwise coverage probabilities of each trajectory in the simulated ODE system with $k = 8$ for 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.

As shown in Table C.2, the trajectory MSE of $k = 8$ is slightly smaller than those of $k = 5$, which is expected as the new variables are linear in time. In terms of the

MSE for estimating the functions, we can expect the performance to be similar, since the non-zero functions are identical in both cases. As $k = 8$ has a higher proportion of vanishing components than $k = 5$, the MSE for all f_{ip} 's will be slightly higher. Regarding system recovery, these two systems do not have a significant difference. Figure C.6 depicts the pointwise coverage probabilities for each trajectory for this larger system, where the performances are similar to the Figure C.1 for the case of $k = 5$ and provide reliable inference for the ODE solutions. In summary, our method works well in a larger system.

C.2.2 Absolute value functions

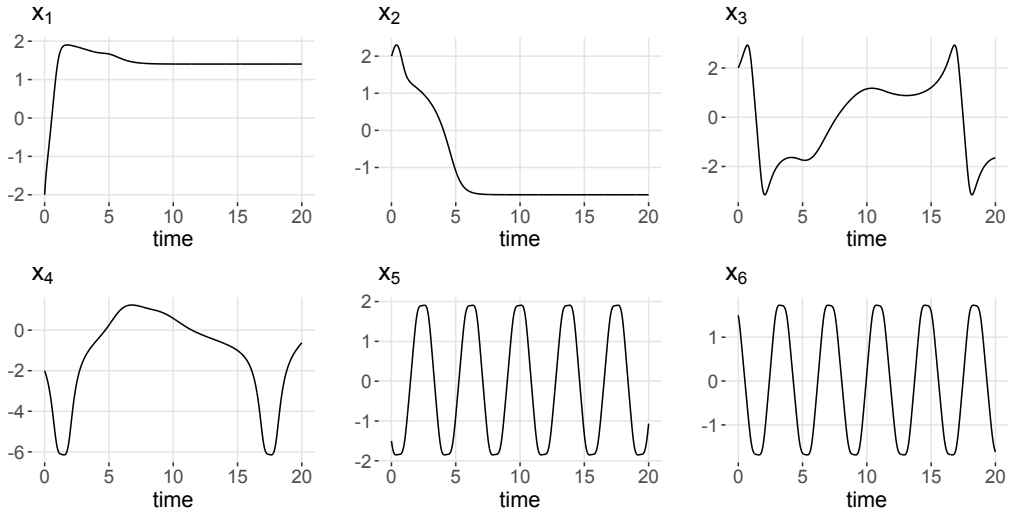


Figure C.7: The true trajectories $\mathbf{x}_1, \dots, \mathbf{x}_6$ of the simulated ODE system (C.1) in the time interval $[0, 20]$.

To assess the robustness of our method when the true functions deviate significantly from those represented by the B-spline approximation, we consider the following system,

$$\left\{ \begin{array}{l} x'_{2k-1}(t) = \beta_{2k-1,0} + \beta_{2k-1,1}^\top \mathbf{g}\{x_{2k-1}(t)\} + \beta_{2k-1,2}^\top \mathbf{g}\{x_{2k}(t)\} \\ x'_{2k}(t) = \beta_{2k,0} + \beta_{2k,1}^\top \mathbf{g}\{x_{2k-1}(t)\} + \beta_{2k,2}^\top \mathbf{g}\{x_{2k}(t)\} \\ x'_{2k-1}(t) = \beta_{2k-1,0} + \beta_{2k-1,1}^\top \mathbf{h}\{x_{2k-1}(t)\} + \beta_{2k-1,2}^\top \mathbf{h}\{x_{2k}(t)\} \\ x'_{2k}(t) = \beta_{2k,0} + \beta_{2k,1}^\top \mathbf{h}\{x_{2k-1}(t)\} + \beta_{2k,2}^\top \mathbf{h}\{x_{2k}(t)\} \end{array} \right. , \text{ if } k = 2 \quad \text{, otherwise} \quad (C.1)$$

where $k = 1, \dots, 5$, and $\mathbf{g}(x) = (|x|, x^2, x^3)$ and $\mathbf{h}(x) = (x, x^2, x^3)$. Compared to the simulation system (17) in Section 3.1 of the main text, when $k = 2$, $\mathbf{g}(x)$ contains an

absolute value function that cannot be approximated by B-splines. As in Section 3.1, the initial values $x_i(0)$ for $i = 1, \dots, 10$ and the parameters $\beta_{i,m}$ for $m = 0, 1, 2$ remain unchanged. The true trajectories $\mathbf{x}_1, \dots, \mathbf{x}_6$ are shown in Figure C.7 to illustrate the differences between the two systems. We generate 41 equally spaced observations over the time interval $[0, 20]$ with a noise level of $\text{SNR} = 25$, and set $\eta_1 = 20$ and $\eta_0 = 0.25$. All other settings, including the implementation of our method and competing methods, are identical to those used for the 41-observation case with $\text{SNR} = 25$ in the main text. Similarly, 100 independent simulation trials are conducted to evaluate the performance of the estimation.

The results for trajectory recovery, estimation of ODE additive components, and network discovery are summarized in Table C.3. The pattern mirrors that observed in simulation system (17) in the main text: although our method does not achieve the highest accuracy in identifying the true additive components, it outperforms the competitors by attaining the lowest MSE for the additive components while maintaining a fairly high TPR. Due to rapid changes in the true system, all three methods yield slightly higher trajectory MSEs compared to the case without absolute value functions. Nevertheless, all methods recover the trajectories reasonably well.

Table C.3: Comparison of methods for system (C.1) with 41 observations and $\text{SNR} = 25$ in trajectory recovery, ODE additive components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.

Method	$\text{MSE}(\hat{\mathbf{x}})$ ($\times 10^{-2}$)	$\text{MSE}_{\mathcal{A}}(\hat{f})$	$\text{MSE}_{\mathcal{A}^c}(\hat{f})$ ($\times 10^{-2}$)	$\text{MSE}(\hat{f})$	TPR	FPR
Integral	0.374 (1.461)	0.102 (0.077)	0.057 (0.313)	0.009 (0.009)	0.973 (0.079)	0.192 (0.054)
JADE	0.128 (0.050)	0.126 (0.104)	0.463 (2.300)	0.014 (0.025)	0.888 (0.101)	0.047 (0.033)
GRADE	0.271 (0.025)	0.216 (0.030)	0.545 (0.945)	0.022 (0.010)	1.000 (0.000)	0.104 (0.053)

The pointwise coverage probabilities for each trajectory are shown in Figure C.8. The results for $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_5$ through $\mathbf{x}_{10}$ are similar to those in the case without the absolute value function. Since $\mathbf{x}_3$ and $\mathbf{x}_4$ exhibit rapid changes, the coverage probabilities can be expected to fall below the nominal level at those time points. Overall, our method provides reasonable inference for the ODE solutions. Consequently, these results demonstrate that

our method can effectively handle cases where the true functions deviate significantly from those represented by the B-spline approximation.

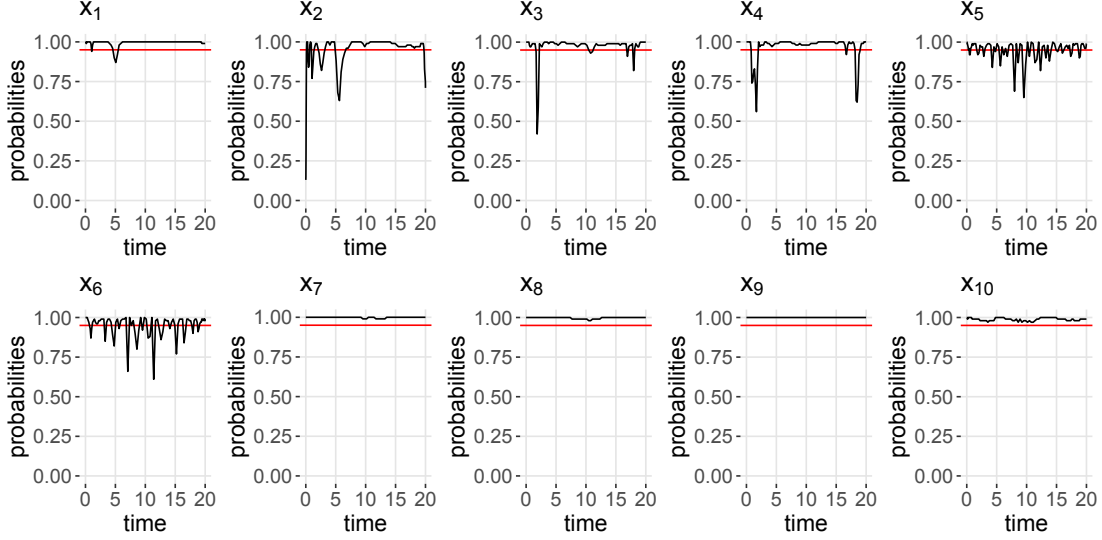


Figure C.8: The pointwise coverage probabilities of each trajectory in the simulated ODE system (C.1) for 41 observations over 100 simulation runs when SNR=25 (black line). The red horizontal line indicates the nominal level 95%.

C.2.3 Lotka-Volterra equations

To investigate the performance of our method in ODE systems with interactions, we consider the Lotka–Volterra (LV) equations (Lotka, 1932) which describes the interaction between the populations of prey (x_{2k-1}) and predator (x_{2k}) over time,

$$\begin{cases} x'_{2k-1}(t) = \beta_{2k-1,1}x_{2k-1}(t) - \beta_{2k-1,2}x_{2k-1}(t)x_{2k}(t) \\ x'_{2k}(t) = \beta_{2k,1}x_{2k-1}(t)x_{2k}(t) - \beta_{2k,2}x_{2k}(t) \end{cases}, \quad (\text{C.2})$$

where $k = 1, \dots, 5$, $\beta_{2k-1,1} = 1.1 + 0.2(k-1)$, $\beta_{2k-1,2} = 0.4 + 0.2(k-1)$, $\beta_{2k,1} = 0.1 + 0.2(k-1)$, and $\beta_{2k,2} = 0.4 + 0.2(k-1)$. The initial values are set as $\{x_{2k-1}(0)\}_{k=1,\dots,5} = (5, 4, 7.5, 5.5, 8)$, with $x_{2k-1}(0) = x_{2k}(0)$. We generate 41 equally spaced observations of x_i , for $i = 1, \dots, 10$, over the time interval $[0, 20]$, with a noise level of SNR = 25. We place 81 knots equally spaced on $[0, 20]$ for $\Phi(t)$, and one interior knots on $\hat{\mathbf{x}}(t)$ without an intercept for each basis function $\Psi(\hat{\mathbf{x}}(t))$. Regarding the quadrature points, we choose $M_i = 200$ to estimate

the outer integral (16), following the guideline in Section 2.2. We fix $\lambda = 10$ since the observations are relatively sparse for this simulated data. As recommended by [Alhamzawi and Ali \(2018\)](#), the hyperparameters in the prior of θ are set as $\alpha_\theta = \beta_\theta = 1$. For the parameters in the prior of $\mathbf{a}_{ip}$, we set $\eta_1 = 5$ and $\eta_0 = 0.5$.

As in the performance evaluation of the simulation studies described in Section 3.1 of the main text, our proposed method is compared with JADE and GRADE. For JADE, a maximum of four iterations is used in their block descent algorithm to update the parameters, and a cubic spline basis with four knots is employed for the spline approximation of f_{ip} ; for GRADE, we use the default configurations. For the parameter estimates, we take the sample mean for the two competing methods, while our method uses the posterior mean of the sample. To assess estimation performance, 100 independent simulation trials are conducted. We use the R package *deSolve* ([Soetaert et al., 2010](#)) to numerically solve the differential equations for generating the simulation data.

The results of the mean and standard error of each performance metric are presented in Table C.4. Since all three methods assume additive ODE models, we are unable to calculate $\text{MSE}_{\mathcal{A}}(\hat{f})$ and $\text{MSE}(\hat{f})$ for this Lotka–Volterra equations (C.2) that include interactions. Despite that our method may not achieve the highest TPR and the lowest FPR, it can identify both zero and non-zero components at an acceptable rate. In contrast, JADE provides the lowest FPR but also the lowest TPR, while GRADE achieves the highest TPR but a significantly high FPR. When estimating the vanishing components, our method outperforms the two competitors with the lowest MSE. Although GRADE attains the lowest MSE for trajectory estimation, it fails to estimate the zero components accurately, as indicated by a dramatically high $\text{MSE}_{\mathcal{A}^c}(\hat{f})$. Overall, our method demonstrates competitive performance of systems with interactions in terms of trajectory estimation, vanishing component estimation and network discovery.

Table C.4: Comparison of methods for Lotka-Volterra equations with 41 observations and $\text{SNR} = 25$ in trajectory recovery, ODE vanishing components estimation and network discovery. Standard errors are displayed in the parentheses under the evaluation scores.

Method	$\text{MSE}(\hat{\mathbf{x}})$	$\text{MSE}_{\mathcal{A}^c}(\hat{f})$	TPR	FPR
Integral	0.499 (0.045)	0.136 (0.097)	0.825 (0.064)	0.271 (0.043)
JADE	0.882 (0.064)	0.381 (0.451)	0.392 (0.044)	0.065 (0.035)
GRADE	0.205 (0.003)	4.423 (1.337)	0.969 (0.032)	0.938 (0.071)

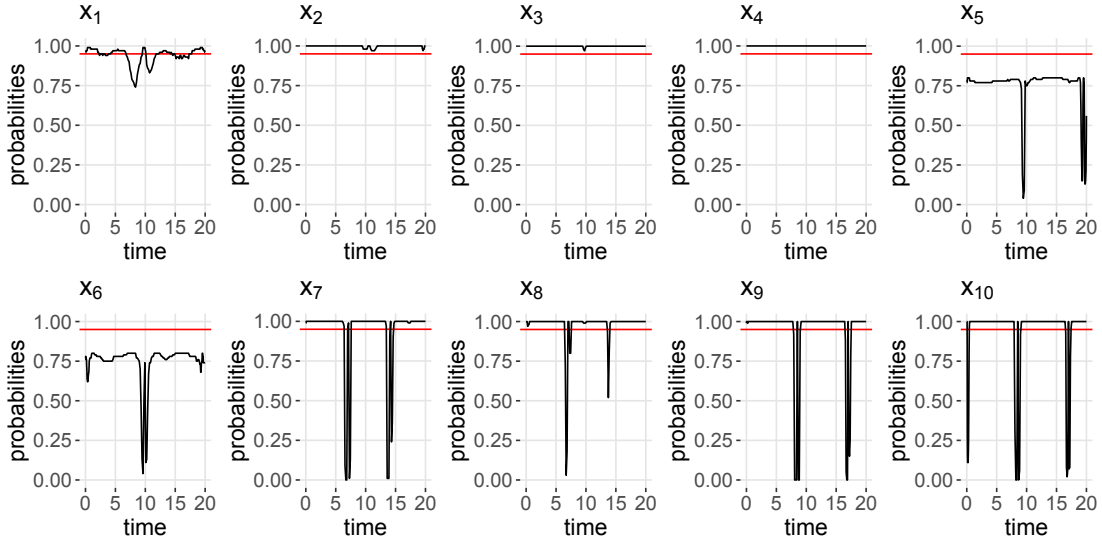


Figure C.9: The pointwise coverage probabilities of each trajectory in the simulated ODE system Lotka-Volterra equations with 41 observations over 100 simulation runs when $\text{SNR}=25$ (black line). The red horizontal line indicates the nominal level 95%.

Figure C.9 shows the pointwise coverage probabilities for each trajectory. Since the observations are relatively sparse within the time interval, we can expect that the estimation of the trajectories can be slightly inaccurate; as a result, the pointwise coverage probabilities of the trajectories may fall below the nominal level 95%, especially for trajectories $\mathbf{x}_5$ and $\mathbf{x}_6$. On the other hand, since the noise level is relatively low, the pointwise coverage

probabilities of the trajectories can be close to 100%, particularly for trajectories $\mathbf{x}_2$ to $\mathbf{x}_4$, which are smoother than the other trajectories. Nevertheless, our method can provide reasonable inference for the ODE solution for systems with interactions.

C.3 Real data analysis: additional results

Table C.5 presents the variable selection results of T-cell activation GRN from our method, which details the inward and outward influence genes identified for each gene. For example, FYB (gene 45) has gene 13 as its inward influence gene, meaning gene 13 regulates FYB. Alternatively, gene 6 is one of outward influence genes of FYB, meaning FYB regulates gene 6.

To assess the sensitivity to priors, we run 10 independent trials for real data with five combinations of hyperparameter $\boldsymbol{\eta}$ values at a fixed rate of $\eta_1/\eta_0 = 10/0.6$. All other settings remain the same as in Section 3.4. For these different $\boldsymbol{\eta}$ values, the computation costs are only slightly different and the average computation time is approximately 15.92 hours. Out of 3364 possible functions, roughly 150-400 are chosen to be non-zero for all these $\boldsymbol{\eta}$ values and each gene is only regulated by a few other genes, which are consistent with our results in Section 3.4 of the main text.

Table C.6 presents the top 15 genes with the highest numbers of outward connections with different $\boldsymbol{\eta}$ values. Among 58 genes, only 41 genes have been selected at least once, which confirms the sparseness of our reconstructed network. Based on 50 independent trials, FYB (gene 45) has been chosen 49 times with the highest frequency as being one of the most important genes. This result is in accordance with our results in Section 3.4 of the main text. The frequency of each gene selected to be an outward connection of FYB over 50 trials is shown in Table C.7. In line with the findings in Wu et al. (2014), genes 7, 15, 18 and 55 are regulated by FYB with moderately high frequencies.

Despite the importance of FYB as a gene in T-cell activation with a high number of outward influence genes, it can have a wide variety of different outward connections. The reason may be that some direct connections we discovered are mediated by genes not included in our data, and such connections might not be strong enough given this dataset. Additionally, Rangel et al. (2005) demonstrated that the fidelity of network reconstruction can be improved with data containing more replicates and additional time points.

Gene	Inward influence genes	Outward influence genes
1	10, 26, 31, 41, 44, 52, 56	55
2	5, 23, 28	
3	23, 29, 32	5, 43
4	25, 28	32, 49, 56
5	3, 5, 9, 14, 19, 24, 31, 40, 47, 49, 50, 53	2, 5, 10, 15, 18, 22, 25, 32, 36, 37, 48, 53, 54, 56
6	7, 28, 29, 45, 58	29
7	13, 36, 45, 56	6, 22, 28, 38, 50
8	9, 37, 43, 58	33
9	11	5, 8, 20, 21, 22
10	5, 11, 16, 18, 26, 28, 36, 41, 46, 56	1, 19
11	23, 28, 33, 57	9, 10, 13, 18, 20, 21, 23, 24, 32, 36, 38, 39, 42, 47, 48, 52, 56
12	31, 32	43
13	11, 32	7, 15, 24, 40, 44, 45, 49, 55
14	57	5, 24, 33, 39
15	5, 13, 21, 31, 33, 45	22, 26
16	19, 36, 52, 56	10, 20, 21, 26, 27, 30, 32, 33, 52
17	21, 29, 31	33
18	5, 11, 40, 57	10
19	10, 19	5, 16, 19, 22, 36, 55
20	9, 11, 16, 20, 29, 34, 40, 48, 56, 57	20, 25, 29, 43, 47
21	9, 11, 16, 39, 52	15, 17, 27, 43
22	5, 7, 9, 15, 19, 25, 29, 34, 42, 43, 47, 51, 52, 55, 57, 58	27, 38, 49
23	11, 23, 25, 40, 45, 52	2, 3, 11, 23, 27, 28, 29, 32, 34, 36, 38, 55, 58
24	11, 13, 14, 30, 32	5
25	5, 20, 28, 39	4, 22, 23, 27, 55
26	15, 16, 40, 55	1, 10, 32, 35
27	16, 21, 22, 23, 25, 29, 32, 33, 48, 50	
28	7, 23, 29, 40, 49, 56	2, 4, 6, 10, 11, 25, 33, 34, 40, 43, 47, 52, 54, 55
29	6, 20, 23	3, 6, 17, 20, 22, 27, 28, 34, 36, 38, 43, 48, 49, 52, 56
30	16, 32, 42, 46, 56	24, 42, 43, 50
31		1, 5, 12, 15, 17
32	4, 5, 11, 16, 23, 26, 32, 38, 50, 56, 58	3, 12, 13, 24, 27, 30, 32, 33, 47
33	8, 14, 16, 17, 28, 32, 48, 51	11, 15, 27, 37, 44, 52, 58
34	23, 28, 29	20, 22
35	26	49
36	5, 11, 19, 23, 29, 51	7, 10, 16, 44, 46, 53
37	5, 33, 45	8, 46, 55
38	7, 11, 22, 23, 29, 39, 57	32
39	11, 14	21, 25, 38, 56
40	13, 28	5, 18, 20, 23, 26, 28, 47, 49, 53, 54
41		1, 10
42	11, 30	22, 30
43	3, 12, 20, 21, 28, 29, 30, 52	8, 22, 54, 55
44	13, 33, 36, 52	1
45	13	6, 7, 15, 23, 37, 46, 49, 50, 52, 55
46	36, 37, 45	10, 30, 57
47	11, 20, 28, 32, 40	5, 22
48	5, 11, 29, 52	20, 27, 33, 54
49	4, 13, 22, 29, 35, 40, 45, 51, 58	5, 28, 50
50	7, 30, 45, 49	5, 27, 32, 52
51		22, 33, 36, 49, 55
52	11, 16, 28, 29, 33, 45, 50	1, 16, 21, 22, 23, 43, 44, 48, 55
53	5, 36, 40, 56	5
54	5, 28, 40, 43, 48, 55, 58	
55	1, 13, 19, 23, 25, 28, 37, 43, 45, 51, 52	22, 26, 54
56	4, 5, 11, 29, 39	1, 7, 10, 16, 20, 28, 30, 32, 53
57	46	11, 14, 18, 20, 22, 38
58	23, 33	6, 8, 22, 32, 49, 54

Table C.5: Variable selection results of T-cell activation gene regulation network.

Gene	Frequency	Gene	Frequency	Gene	Frequency
12	1	48	8	31	29
34	1	26	10	40	29
42	1	58	10	13	35
55	1	5	11	25	37
43	2	16	12	30	38
15	3	51	12	29	41
27	3	39	13	57	41
10	4	49	13	11	47
19	4	52	18	28	47
14	5	50	20	23	48
54	5	37	21	45	49
56	5	32	26		
20	6	33	27		
46	6	36	27		
47	6	7	28		

Table C.6: Frequency of top 15 genes with highest number of outward influence genes over 50 independent trials in T-cell activation gene regulation network with different $\boldsymbol{\eta}$ values.

Gene	Frequency	Gene	Frequency
35	3	28	11
1	4	32	11
2	5	36	11
37	5	38	11
52	5	51	11
6	6	8	12
31	6	10	12
53	6	20	12
57	6	54	12
24	7	9	13
26	7	27	13
39	7	47	13
40	7	49	13
44	7	4	14
33	8	16	14
46	8	17	14
13	9	3	15
19	9	7	15
21	9	18	15
23	9	25	16
50	9	58	16
56	9	34	17
14	10	29	18
41	10	22	19
42	10	11	20
45	10	15	20
48	10	30	20
5	11	55	21
12	11	43	23

Table C.7: Frequency of each gene being an outward influence gene of FYB (gene 45) over 50 independent trials in T-cell activation gene regulation network with different $\boldsymbol{\eta}$.

Appendix D

Additional Results for Chapter 4

D.1 Simulation studies: additional results

D.1.1 Sensitivity of τ

To assess the sensitivity of τ , we consider using 2τ , 5τ , 0.5τ and 0.2τ when we sample θ , $\mathbf{c}$, $\mathbf{e}$ and σ from the posterior distribution $\pi(\theta, \mathbf{c}, \mathbf{e}, \sigma | \mathbf{y}, \lambda, \tau)$ given by (4.6). Figure D.1 depicts the average estimated inadequacy function for τ , 2τ and 5τ across 100 simulation runs, along with the corresponding central 95% intervals, while Figure D.2 shows the average estimated inadequacy function for τ , 0.5τ and 0.2τ . As τ increases, we can expect the estimated inadequacy functions to slightly deviate more from zero line, and their corresponding 95% intervals to become wider; as τ decreases, the estimated inadequacy functions may shrink toward zero line, and the corresponding 95% intervals will be narrower. Despite this, our proposed method can still provide reasonable inferences for both models, as the value set for τ has only a minor impact on the estimation of the inadequacy function. Regarding model validation, Model 1 is treated as valid across 100 simulation runs for different values of τ , while Model 2 is consistently treated as invalid across 100 simulation runs. Table D.1 presents the results for parameter estimation in Model 1 using our proposed method across 100 simulation runs, where the mean and RMSE show only slight differences across different values of τ . Overall, our method provides reliable inferences for validating the models and parameter estimates and is not highly sensitive to the value set for τ .

Hyperparameter	θ_1	θ_2	θ_3	θ_4
τ	0.500(0.010)	0.500(0.012)	1.010(0.049)	1.011(0.050)
2τ	0.500(0.010)	0.500(0.012)	1.010(0.049)	1.011(0.050)
5τ	0.500(0.010)	0.500(0.012)	1.010(0.048)	1.011(0.050)
0.5τ	0.500(0.010)	0.500(0.012)	1.010(0.049)	1.011(0.050)
0.2τ	0.499(0.010)	0.499(0.012)	1.010(0.047)	1.011(0.049)

Table D.1: Mean of the parameter estimates with RMSE in parentheses for Model 1 (4.7) under different τ across 100 simulated runs.

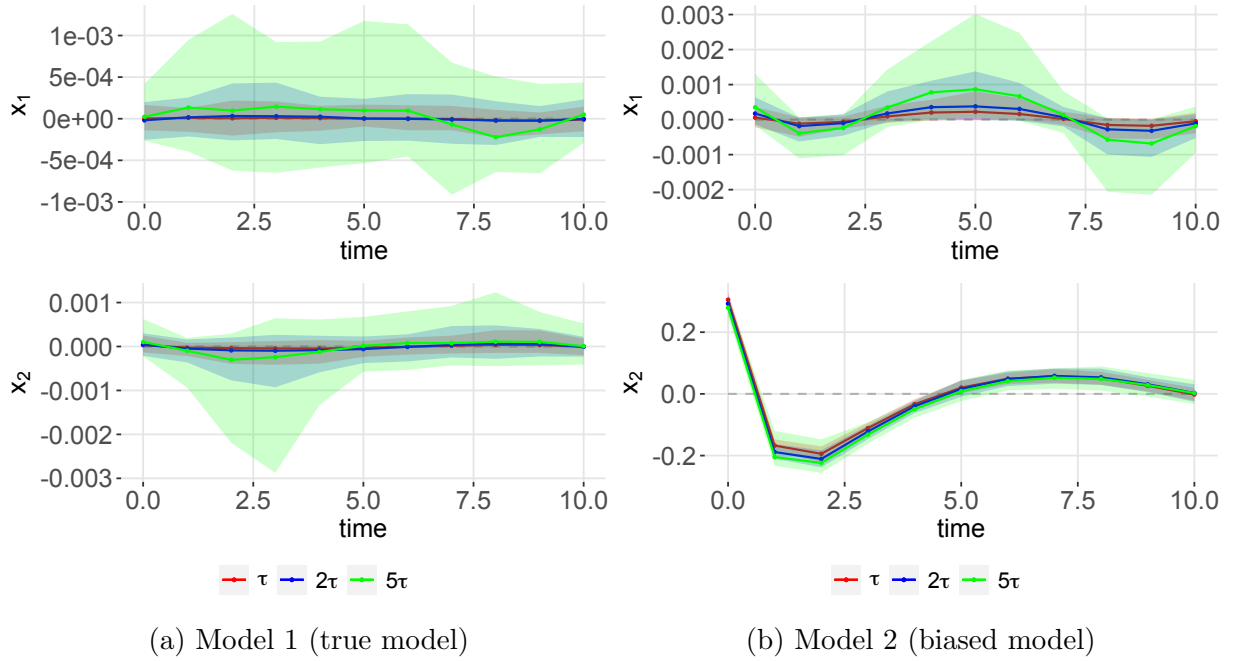


Figure D.1: Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with low-noise settings. The black dashed line represents zero line. The red, blue and green solid line represents the average estimated inadequacy function for τ , 2τ , and 5τ , respectively. The red, blue and green shaded area represents the corresponding pointwise central 95% intervals of all estimated inadequacy functions for τ , 2τ , and 5τ , respectively.

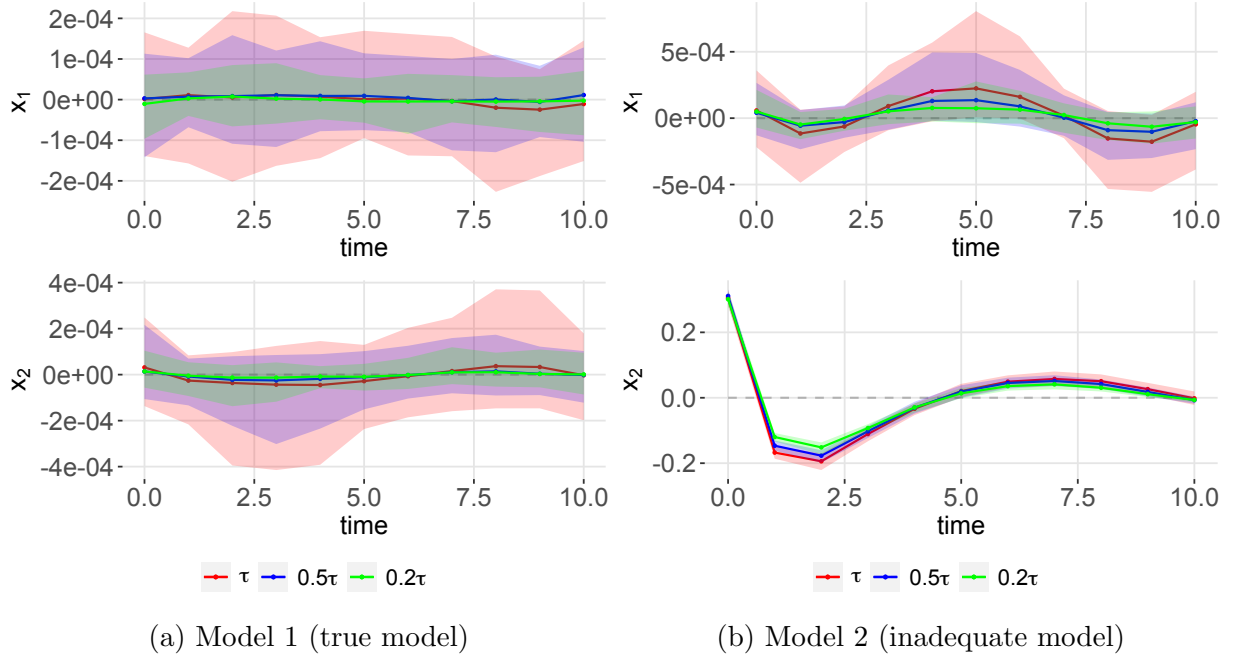


Figure D.2: Estimated inadequacy function for x_1 (top panel) and x_2 (bottom panel), over 100 simulation runs with low-noise settings. The black dashed line represents zero line. The red, blue and green solid line represents the average estimated inadequacy function for τ , 0.5τ , and 0.2τ , respectively. The red, blue and green shaded area represents the corresponding pointwise central 95% intervals of all estimated inadequacy functions for τ , 0.5τ , and 0.2τ , respectively.