
Likelihood-based computational analysis and uncertainty quantification for mechanistic models

Joel Trent

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Engineering in Engineering Science, the University of Auckland, 2024. This thesis is for examination purposes only and is confidential to the examination process.

Abstract

Uncertainty quantification workflows in science use mathematical models to understand and predict physical systems given observations. Recently, the *Profile-Wise Analysis* (PWA) workflow has been introduced as a likelihood-based workflow for identifiability analysis, parameter estimation and model prediction, which features approximate frequentist coverage guarantees alongside a reduction in the computational cost of using a likelihood-based workflow. We provide a needed quantitative assessment of the PWA workflow in terms of coverage and computational performance (which are not well known) and implement it as an open-source *Julia* package, `LikelihoodBasedProfileWiseAnalysis` to improve its accessibility and availability for use on other models. In doing so, we improve the computational cost of the workflow. We also formally introduce reference tolerance sets for uncertainty quantification of observations and demonstrate their value in contrast to more traditional prediction intervals. Moreover, we show that modifying the degrees of freedom parameter proposed for evaluating univariate and bivariate profiles to be instead equal to the number of model parameters significantly improves the coverage performance of profile-wise prediction sets on well-identified models at no additional computational cost. We present this as ‘profile paths’ for PWA, which trace the boundary of the full likelihood.

In particular, we show that the approximate profile-wise sets for predictive quantities formed using the union of simultaneous bivariate profiles have similar levels of statistical accuracy to prediction sets from full parameter confidence sets (the ‘gold standard’) at up to a 93% decrease in computational cost on the models considered. This computational cost can be further decreased by considering a good subset of these profiles or further improving the optimisation scheme. Unlike other efficient likelihood-based prediction methods, we also show that these profile-wise prediction sets have approximate simultaneous guarantees. The performance of the methods we propose for finding these bivariate profiles is also not strongly dependent on having relatively well-informed parameter bounds. We also find that the $(1 - \delta, 1 - \alpha)$ reference tolerance sets we introduced can be appropriate for trapping at least $1 - \delta$ of observations with confidence $1 - \alpha$ when formed from at least the union of simultaneous univariate profiles. Open source software on Github can be used to replicate our experiments.

Acknowledgements

To my supervisors, Oliver, Ru and Matthew, thank you for your time, interest and knowledge of the thesis topic - a special thank you to Ru and Matthew for covering for Oliver while he was on parental leave. I was able to engage and understand the subject to a large degree because of you all, which I hope is reflected through this thesis.

A thank you to Kevin for always being around to check in with when I came on campus, particularly on weeks my supervisors were on leave. Thank you for your thoughts and time over the last six years.

I would also like to thank my friend Josh, among others, for continuing to be someone with whom I can laugh at the frustrations of creating and discovering bugs in my code. Thank you to my other friends for your interest in how things have been going and what I'm up to; your support has meant a lot.

A big thank you to my family for their continued support over the last year. It means a lot to me, and I would not have been able to achieve what I have during University without you all.

Declaration of Contribution

All work done in this project is solely the work of the author, with the exception of the contributions listed below:

The implementation and parameterisation of the three main models considered in this thesis is based on code provided alongside published works by Simpson and Maclaren [1] and Simpson et al. [2]. The implementation of the PWA workflow as the *Julia* package `LikelihoodBasedProfileWiseAnalysis` [3] was started from these set of scripts.

To follow along with the examples in text see the relevant models in the examples section of `LikelihoodBasedProfileWiseAnalysis`' documentation [3]. This section includes graphs for visualising the various sets. For the full details on constructing all outputs in this thesis, see the *Bespoke graphics* and *Experiments* folders of `UoA_MastersWorking`. Within the *Experiments* folder the *Models* folder contains the implementation of each model and the required functions/variables for `LikelihoodBasedProfileWiseAnalysis`. The *Outputs* folder contains all experiment outputs. `Experiments.ipynb` contains Julia code used to create all figures within this text which are then saved in *Outputs*. The other files in the *Experiments* folder contain the code used to run the experiments on each model. The ‘C?’ prepended to each *Julia* file name gives the section in which the model is first used, e.g. `C3_logistic.jl` refers to the logistic model which is first used in Section 3.

Some aspects of the descriptions of profiling algorithms, the considered models in this text and the workflow formulation in the literature review is taken from the online documentation of the author’s *Julia* packages (or similarly was taken from this thesis for that documentation) `LikelihoodBasedProfileWiseAnalysis` [3], `EllipseSampling` [4] and `UnivariateUnimodalHighestDensityRegion` [5], for which there are no other contributors.

Many open-source *Julia* packages were used in the implementation of the author’s *Julia* packages, as well as the simulation results. These directly include, along with their dependencies, `Accessors`, `AngleBetweenVectors`, `Arrow`, `Clustering`, `Combinatorics`, `ConcaveHull`, `CSV`, `DataFrames`, `DataStructures`, `DifferentialEquations`, `Distances`, `Distributed`, `Distributions`, `Elliptic`, `FLoops`, `ForwardDiff`, `HypothesisTests`, `LaTeXStrings`, `LatinHypercubeSampling`, `LinearAlgebra`, `LogExpFunctions`, `Mesher`, `OptimizationNLopt`, `Optimization`, `Plots`, `PolygonInbounds`, `ProgressMeter`, `Random`, `Reexport`, `Requires`, `Revise`, `Roots`, `SciMLBase`, `SharedArrays`, `StaticArrays`, `StatsBase`, `StatsPlots`, `TimerOutputs`, `TrackingHeaps` and `TravelingSalesmanHeuristics`.

Contents

Abstract	ii
Acknowledgements	iii
Declaration of Contribution	iv
List of Tables	xiii
List of Algorithms	xv
Notation List	xvi
1 Introduction	1
1.1 Project Scope and Research Objectives	2
2 Literature Review	3
2.1 Why Use The PWA Workflow: Types of Intervals and Sets	3
2.1.1 For Parameters	4
2.1.2 Predicting Observations	5
2.2 PWA Workflow Formulation	6
2.2.1 Observed Data	6
2.2.2 Mechanistic Mathematical Model	6
2.2.3 Data Distribution Parameters	7
2.2.4 Likelihood Function	8
2.2.5 Profile Likelihood Function	9
2.2.6 Confidence Sets For Parameters	9
2.2.7 Confidence Sets For Data Distribution Parameters	10
2.2.8 Reference Tolerance Sets For Observed Data	11
2.2.8.1 Full Reference Tolerance Set Coverage	13
2.2.9 Numerical Implementation	14
2.3 PWA Workflow Background and Existing Methods	15
2.3.1 Ellipse/Elliptical Approximation of the Log-Likelihood Function	15
2.3.1.1 Profile Likelihood Approximation	16
2.3.1.2 Improved Approximations	17

2.3.2	Parameter Identifiability and Sloppiness	17
2.3.2.1	Structurally Non-Identifiable Parameters	18
2.3.2.2	Practically Non-Identifiable Parameters	19
2.3.2.3	Identifiable Parameters	19
2.3.2.4	Model Reduction	19
2.3.3	Parameter Confidence Intervals	20
2.3.3.1	Wald Confidence Intervals	20
2.3.3.2	Likelihood-Based Confidence Intervals	21
2.3.3.3	Numerical Implementations	21
2.3.4	Estimating Likelihood-Based Confidence Sets For Parameters	22
2.3.4.1	General approaches	23
2.3.4.2	Univariate Profiles	24
2.3.4.3	Bivariate Profiles	24
2.3.5	Polygons and Point Clouds	24
2.3.5.1	Minimum Perimeter Polygon Problems	24
2.3.5.2	Convex Hulls	25
2.3.5.3	Concave Hulls and Alpha Shapes	25
2.3.5.4	Level Set Continuation, Star Domains and Polygon Pole of Inaccessibility	25
2.3.6	Sets for Predictive Quantities	26
2.3.6.1	Trajectory Confidence Sets	26
2.3.6.2	Observation Reference Tolerance Sets	27
2.3.7	Coverage Testing Confidence Intervals, Sets and Reference Tolerance Sets	29
2.3.7.1	Parameter Confidence Intervals	29
2.3.7.2	Bivariate Profiles	29
2.3.7.3	Full and Profile-Wise Trajectory Confidence Sets	30
2.3.7.4	Full and Profile-Wise Reference Tolerance Sets and Observations	30
2.3.8	Area Coverage Testing of the Polygon Approximation of the True Boundary of a Bivariate Profile	31
3	Computing Univariate Profiles	32
3.1	Parameter Identifiability and Finding Univariate Confidence Set Boundaries	32
3.1.1	Parameter Identifiability	33
3.1.2	Likelihood-based Confidence Intervals and Wald Confidence Intervals	34

3.1.2.1	Ellipse Approximation Consistency: Analytical and Optimised	35
3.1.3	Algorithm Performance	37
3.1.3.1	Bracketing Method Choice	37
3.1.3.2	Impact Of Tolerance Settings	37
3.1.3.3	More Efficiently Finding Confidence Intervals at Multiple Levels . .	40
3.1.3.4	Using Wald Confidence Intervals To Reduce The Search Bracket . . .	41
3.1.3.5	Areas for Improvement	42
3.2	Parameter Confidence Interval Coverage	43
3.2.1	Logistic and Lotka-Volterra Models	44
3.2.2	Two-Species Logistic Model	44
3.2.2.1	Logit-Normal Data Distribution	46
3.3	Performance Comparisons With <code>LikelihoodProfiler</code>	47
3.3.1	Logistic	48
3.3.2	Lotka-Volterra	49
3.3.3	Two-Species Logistic With Logit-Normal Data Distribution	49
4	Efficiently Computing Bivariate Profiles	51
4.1	Finding Profile Boundaries	51
4.1.1	<code>Fix1AxisMethod</code>	51
4.1.2	<code>SimultaneousMethod</code>	53
4.1.3	<code>RadialRandomMethod</code>	54
4.1.3.1	On Logarithmic Reparameterisations	55
4.1.4	<code>AnalyticalEllipseMethod</code> Using <i>Julia</i> Package <code>EllipseSampling</code> . .	56
4.1.4.1	Clustering Parameter Impact	57
4.1.5	<code>ContinuationMethod</code>	58
4.1.6	<code>RadialMLEMethod</code>	59
4.1.7	<code>IterativeBoundaryMethod</code>	60
4.2	Sampling Inside Profile Boundaries	63
4.3	Performance of Methods on Logistic Model	64
4.3.1	Optimisation Function Call Performance	66
4.3.2	Coverage of True Bivariate 95% Confidence Set Boundary	67
4.3.3	Comparing Optimisation Function Call Performance to Rejection Sampling .	69
4.4	Coverage of True Bivariate Boundary Additional Models	71
4.5	Bivariate Parameter Coverage of Profile Boundaries	72

4.5.1	Logistic and Lotka-Volterra Models	73
4.5.2	Two-Species Logistic With Logit-Normal Data Distribution	73
5	Profile-Wise Analysis: Coverage of Predictive Quantities on Simple Models	75
5.1	Full Parameter Vector Likelihood-Based Confidence Sets	76
5.1.1	Full Trajectory Confidence Set Coverage	77
5.1.2	Full Reference Tolerance Set Coverage	79
5.1.3	Observation Coverage	80
5.2	Univariate Profiles	82
5.2.1	Profile-Wise Trajectory Confidence Set Coverage	82
5.2.2	Profile-Wise Reference Tolerance Set Coverage	84
5.2.3	Observation Coverage	86
5.3	Bivariate Profiles	87
5.3.1	Profile-Wise Trajectory Confidence Set Coverage	88
5.3.1.1	Impact of Observations of Important Features	90
5.3.2	Profile-Wise Reference Tolerance Set Coverage	91
5.3.3	Observation Coverage	92
6	Profile Paths For Profile-Wise Analysis: Coverage of Predictive Quantities on Simple Models	94
6.1	Profile Path Example	95
6.2	Simultaneous Univariate Profiles	96
6.2.1	Profile-Wise Trajectory Confidence Set Coverage	97
6.2.2	Profile-Wise Reference Tolerance Set Coverage	99
6.2.3	Observation Coverage	100
6.3	Simultaneous Bivariate Profiles	101
6.3.1	Profile-Wise Trajectory Confidence Set Coverage	102
6.3.1.1	Reducing Computational Cost While Maintaining Coverage	105
6.3.2	Profile-Wise Reference Tolerance Set Coverage	106
6.3.3	Observation Coverage	107
6.3.4	Impact of Using a Subset of Profiles on Coverage	108
6.3.4.1	Profile-Wise Trajectory Confidence Set Coverage on the Logistic and Lotka-Volterra Models	109

6.3.4.2	Profile-Wise Trajectory Confidence Set and Reference Tolerance Set Coverage on the Two-Species Logistic Model	111
6.3.4.3	Potential Methods For Finding Good Subsets	113
6.4	Performance Against Full Parameter Confidence Sets	114
6.4.1	Improving Computational Performance of Profile-Wise Prediction Sets	115
6.4.2	Logistic and Lotka-Volterra Models	117
6.4.3	Two-Species Logistic Model	119
7	Discussion and Conclusions	122
7.1	Univariate Profiles	122
7.1.1	Ellipse Approximations	122
7.1.2	Implementation Validation	122
7.1.3	Implementation Performance Improvements and Comparisons	123
7.2	Bivariate Profiles	123
7.2.1	Performance of Boundary Methods	124
7.2.2	Implementation Validation	125
7.3	Profile-wise Analysis	125
7.3.1	Prediction Sets From Full Parameter Confidence Sets	125
7.3.2	Profile-Wise Prediction Sets From Univariate and Bivariate Profiles	125
7.4	Profile Paths For Profile-Wise Analysis	127
7.4.1	Profile-Wise Prediction Sets From Simultaneous Univariate Profiles	127
7.4.2	Profile-Wise Prediction Sets From Simultaneous Bivariate Profiles	129
7.4.3	Profile-Wise Prediction Sets From a Subset of Simultaneous Bivariate Profiles	130
7.4.4	Performance Against Full Parameter Confidence Sets	131
7.4.5	Prediction Goals	131
7.5	Stochastic Models	132
7.6	Research Objectives	134
7.7	Conclusions	134
References		136
Appendices		144
A Description of the Example Models		144
A.1	Logistic Model	144

A.2	Lotka-Volterra Model	145
A.3	Two-Species Logistic Model	146
A.3.1	With Logit-Normal Data Distribution	147
A.4	Gaussian Approximation of a Binomial Distribution	148
B	Tolerance Set Coverage Example With Univariate Profiles	149
C	Bracketing Method Choice	150
D	Univariate Profiles	151
D.1	Lotka-Volterra	151
D.2	Two-Species Logistic	151
D.2.1	Logit-Normal Data Distribution	153
E	Bivariate Confidence Boundary Algorithm Pseudocode	155
E.1	General Algorithm used by <code>SimultaneousMethod</code> , <code>RadialRandomMethod</code> and <code>RadialMLEMethod</code>	155
E.1.1	<code>SimultaneousMethod</code> Point Pairs	156
E.1.2	<code>RadialRandomMethod</code> Point Pairs	157
E.1.3	<code>RadialMLEMethod</code> Point Pairs	158
E.1.4	<code>IterativeBoundaryMethod</code> Point Pairs	159
F	Iterative Boundary Further Discussion	163
F.1	Search Direction For External Candidate Points	163
F.2	Relative Magnitude Scaling	164
F.3	Demonstration on a Concave Example	165
G	Bivariate Method Performance	167
G.1	Logistic	167
G.1.1	Optimisation and Likelihood Function Call Performance	167
G.1.2	Coverage of True Bivariate 95% Confidence Set Boundary Using a Convex Hull	168
H	Bivariate Profiles	170
H.1	Lotka-Volterra	170
H.2	Two-Species Logistic	171
I	Reference Tolerance Set Coverage	172

J Full Trajectory Confidence Set Coverage vs Full Reference Tolerance Set Coverage When θ^0 Is Estimated	173
K Profile Computational Performance Used for Final Comparisons	176
K.1 Univariate Profiles	176
K.2 Bivariate Profiles	177

List of Tables

1	The mean likelihood and optimisation rank, likelihood calls, optimisation calls and asymptotic efficiency of bracketing methods in <code>Roots</code> on the three-parameter logistic model [1], when finding the 95% confidence interval for each parameter. A method's rank is calculated per parameter by considering the number of likelihood calls, with ties allowed. The optimisation rank is calculated in the same fashion. See Appendix C for more detail.	38
2	Comparison of the performance of <code>LikelihoodProfiler</code> [22] and the method established in this section on the logistic model [1]. Compares the accuracy of finding parameter confidence intervals on a single data realisation and the mean likelihood function calls (3 s.f.) required across 100 data realisations.	49
3	Comparison of the performance of <code>LikelihoodProfiler</code> [22] and the method established in this section on the Lotka-Volterra model [1]. Compares the accuracy of finding parameter confidence intervals on a single data realisation and the mean likelihood function calls (3 s.f.) required across 100 data realisations.	49
4	Comparison of the performance of <code>LikelihoodProfiler</code> [22] and the method established in this section on the two-species logistic model with a logit-normal data distribution [2]. Compares the accuracy of finding parameter confidence intervals on a single data realisation and the mean likelihood function calls (3 s.f.) required across 100 data realisations.	50
5	Settings used for performance comparisons between bivariate methods.	65
6	Settings used for <code>IterativeBoundaryMethod</code> on each model.	72
7	Profile-wise reference tolerance set coverage of the union of univariate and bivariate profiles evaluated at the univariate and simultaneous asymptotic thresholds in the Gaussian noise model from 2000 coverage simulations. 95% confidence intervals for the simulated coverage are bracketed.	96
8	Likelihood and optimisation call rank, likelihood calls, optimisation calls of bracketing methods in <code>Roots</code> on each parameter in the three-parameter logistic model [1]. The likelihood rank of a method is calculated on a per-parameter basis by considering the number of likelihood calls, with ties allowed. The optimisation rank is calculated in the same fashion. Corresponds to Table 1.	150

9	Performance of the method established in Section 3 in terms of the mean likelihood function calls (3 s.f.) required to find the confidence interval for each parameter across 100 data realisations on the logistic model [1]. $\nu = 1$ is at the univariate threshold, while $\nu = \theta $ is at the simultaneous threshold.	176
10	Performance of the method established in Section 3 in terms of the mean likelihood function calls (3 s.f.) required to find the confidence interval for each parameter across 100 data realisations on the Lotka-Volterra model [1]. $\nu = 1$ is at the univariate threshold, while $\nu = \theta $ is at the simultaneous threshold.	176
11	Performance of the method established in Section 3 in terms of the mean likelihood function calls (3 s.f.) required to find the confidence interval for each parameter across 100 data realisations on the two-species logistic model with a logit-normal data distribution [2]. $\nu = 1$ is at the univariate threshold, while $\nu = \theta $ is at the simultaneous threshold.	177
12	Performance of the method established in Section 4.5 in terms of the mean likelihood function calls (3 s.f.) required to find a specified number of points on the bivariate boundary across 100 data realisations on the logistic model [1]. $\nu = 2$ is at the bivariate threshold, while $\nu = \theta $ is at the simultaneous threshold.	177
13	Performance of the method established in Section 4.5 in terms of the mean likelihood function calls (3 s.f.) required to find a specified number of points on the bivariate boundary across 100 data realisations on the Lotka-Volterra model [1]. $\nu = 2$ is at the bivariate threshold, while $\nu = \theta $ is at the simultaneous threshold.	178
14	Performance of the method established in Section 4.5 in terms of the mean likelihood function calls (3 s.f.) required to find a specified number of points on the bivariate boundary across 100 data realisations on the two-species logistic model with a logit-normal data distribution [2]. $\nu = 2$ is at the bivariate threshold, while $\nu = \theta $ is at the simultaneous threshold.	179

List of Algorithms

1	Bracketing method for the boundaries of the univariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$, of parameter j , θ_j , which has a continuous unimodal profile log-likelihood function. Boundaries correspond to the likelihood-based confidence interval for θ_j . This example sets the confidence level of interest to 95%. Values of ω are the optimal values of the nuisance parameters, $\omega^*(\psi)$, given the most recent evaluation of the profile log-likelihood function.	33
2	Fix1AxisMethod for the boundaries of the bivariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$, of parameters j and k , $\theta_j \cup \theta_k$, which has a continuous unimodal profile log-likelihood function. Algorithm from Simpson and Maclaren [1]. This example sets the confidence level of interest to 95%. Values of ω are the optimal values of the nuisance parameters, $\omega^*(\psi)$, given the most recent evaluation of the profile log-likelihood function. find_zero searches for a zero along the 1D line between the two 2D points ψ_1 and ψ_2 (the method to do this is an implementation detail).	52
3	General method for finding boundaries of the bivariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$, of parameters j and k , $\theta_j \cup \theta_k$, which has a continuous unimodal profile log-likelihood function. The point pairs are found using the point pair algorithm for either SimultaneousMethod (Algorithm 4), RadialRandomMethod (Algorithm 5) or RadialMLEMethod (Algorithm 7). This example sets the confidence level of interest to 95%. Values of ω are the optimal values of the nuisance parameters, $\omega^*(\psi)$, given the most recent evaluation of the profile log-likelihood function. find_zero searches for a zero along the 1D line between the two 2D points ψ_1 and ψ_2 (the method to do this is an implementation detail).	155
4	SimultaneousMethod used to find point pairs for Algorithm 3.	156
5	RadialRandomMethod used to find point pairs for Algorithm 3.	157
6	Generation of <code>num_radial_directions</code> radial directions to search in for Algorithm 5. Converts radial directions to vector directions.	157
7	RadialMLEMethod used to find point pairs for Algorithm 3.	158
8	Sample of <code>totalPoints</code> , <code>ellipsePoints</code> on the ellipse approximation of $\hat{\ell}_p(\psi; y_{1:I}^o)$ for Algorithm 7, using <i>Julia</i> package <code>EllipseSampling</code> [4].	158
9	IterativeBoundaryMethod used to iteratively improve an initial polygon approximation of a bivariate confidence boundary.	159

10	Internal angle objective. Uses <i>Julia</i> package <code>AngleBetweenVectors</code> [125] to calculate angles between adjacent edges (connected by a vertex). If two adjacent edges form a straight boundary then the objective is 0.0 radians, whereas if the boundary has an internal angle of $\pi \div 4$ radians (45 deg) the objective is $\pi \times 3 \div 4$ (135 deg). . .	160
11	Edge length objective.	160
12	Finding ψ_{new} given candidate point ψ_{guess}	161
13	Boundary updates given successfully finding or failing to find ψ_{new}	162

Notation List

I	Total number of distinct data observations where $i = 1, 2, 3, \dots, I$
t_i	Discrete time points of distinct data observations with $i \in [1 : I]$
$t_{1:I}$	Vector of discrete time points of data observations t_i
y_i^o	Observed data at time point t_i
$y_{1:I}^o$	Vector of observed data corresponding to time points $t_{1:I}$
θ^M	A vector of mechanistic model parameters
$z(\theta^M)$	Deterministic mechanistic model solution (trajectory) evaluated with parameters θ^M
$z_i(\theta^M)$	Deterministic mechanistic model solution (trajectory) evaluated with parameters θ^M at time point t_i
θ^o	A vector of any additional observation parameters
θ	The full parameter vector, $\theta = (\theta^M, \theta^o)$
$ \theta $	The total number of parameters
θ_j	Parameter j
a and b	Vectors of lower and upper bounds, respectively, on parameters θ such that $\theta_j \in [a_j, b_j]$
$\phi(\theta)$	A vector of data distribution parameters, depending on θ For data distributions from mechanistic model + error model, $\phi(\theta) = (z(\theta^M), \theta^o)$
$\phi_i(\theta)$	A vector of data distribution parameters at time point t_i For data distributions from mechanistic model + error model, $\phi_i(\theta) = (z_i(\theta^M), \theta^o)$, where the error model is dependent on the mechanistic model via the mean
$p(y; \theta)$	Density function for observed data dependent on model parameters θ and in particular, $\phi(\theta)$
$p(y_i; \theta)$	Density function for observed data dependent on $\phi_i(\theta)$ at time point t_i
$\mathcal{L}(\theta; y_{1:I}^o)$	Likelihood function at parameters θ given observed data $y_{1:I}^o$
$\ell(\theta; y_{1:I}^o)$	Log-likelihood function at parameters θ given observed data $y_{1:I}^o$
$\hat{\theta}$	The maximum likelihood estimate of the parameters θ : values that maximise $\ell(\theta; y_{1:I}^o)$
$\hat{\mathcal{L}}(\theta; y_{1:I}^o)$	Normalised likelihood function, $\hat{\mathcal{L}}(\theta; y_{1:I}^o) = \mathcal{L}(\theta; y_{1:I}^o) \div \mathcal{L}(\hat{\theta}; y_{1:I}^o)$
$\hat{\ell}(\theta; y_{1:I}^o)$	Normalised log-likelihood function, $\hat{\ell}(\theta; y_{1:I}^o) = \ell(\theta; y_{1:I}^o) - \ell(\hat{\theta}; y_{1:I}^o)$

ψ	Interest parameters in θ used for profile likelihood evaluation, where $\theta = (\psi, \omega)$
ω	Nuisance parameters in θ optimised out in profile likelihood evaluation, where $\theta = (\psi, \omega)$. Their optimal values are referred to as $\omega^*(\psi)$
$\hat{\ell}_p(\psi; y_{1:I}^o)$	Normalised profile log-likelihood function for interest parameters ψ given observed data $y_{1:I}^o$
$1 - \alpha$	Confidence level / coverage level of interest (often 95%, with $\alpha = 0.05$)
ν	Degrees of freedom. Typically, $\nu = \theta $ for confidence sets of the full parameter vector and $\nu = \psi $ for profile likelihood-based confidence sets
$\Delta_{\nu, 1-\alpha}$	The $1 - \alpha$ quantile of the χ^2 distribution with ν degrees of freedom
$\ell_c = -\Delta_{\nu, 1-\alpha}/2$	Asymptotic threshold at confidence level $1 - \alpha$ and degrees of freedom ν
$\mathcal{C}_{\theta, 1-\alpha}(y_{1:I}^o)$	Approximate likelihood-based confidence set for θ at confidence level $1 - \alpha$
$\mathcal{C}_{\phi, 1-\alpha}(y_{1:I}^o)$	Approximate likelihood-based confidence set for data distribution parameters, $\phi(\theta)$, at confidence level $1 - \alpha$. From propagating forward $\mathcal{C}_{\theta, 1-\alpha}(y_{1:I}^o)$
$\mathcal{A}_{y, 1-\delta}^\phi(y_{1:I}^o)$	Reference set for data observations, y , containing $1 - \delta$ of the population associated with the single data distribution with parameters ϕ
$\mathcal{C}_{y, (1-\delta, 1-\alpha)}(y_{1:I}^o)$	Approximate likelihood-based reference tolerance set for the $1 - \delta$ population reference set with confidence level $1 - \alpha$ from the union of $\mathcal{A}_{y, 1-\delta}^\phi(y_{1:I}^o)$
$\mathcal{C}_{\psi, 1-\alpha}(y_{1:I}^o)$	Approximate profile likelihood-based confidence set for ψ at confidence level $1 - \alpha$, with corresponding nuisance parameters $\omega^*(\psi)$ set to their optimal value at each ψ
$\mathcal{C}_{\phi, 1-\alpha}^\psi(y_{1:I}^o)$	Approximate profile likelihood-based confidence set for data distribution parameters, $\phi(\psi, \omega^*(\psi))$, at confidence level $1 - \alpha$. From propagating forward $\mathcal{C}_{\psi, 1-\alpha}(y_{1:I}^o)$
$\mathcal{A}_{y, 1-\delta}^{\phi(\psi, \omega^*(\psi))}(y_{1:I}^o)$	Reference set for data observations, y , containing $1 - \delta$ of the population associated with the single data distribution with parameters $\phi(\psi, \omega^*(\psi))$
$\mathcal{C}_{y, (1-\delta, 1-\alpha)}^\psi(y_{1:I}^o)$	Approximate profile likelihood-based reference tolerance set for the $1 - \delta$ population reference set with confidence level $1 - \alpha$ from the union of $\mathcal{A}_{y, 1-\delta}^{\phi(\psi, \omega^*(\psi))}(y_{1:I}^o)$
$\mathcal{H}(\hat{\theta})$	Hessian matrix of the log-likelihood function evaluated at $\hat{\theta}$
$\Gamma(\hat{\theta})$	Matrix pseudo-inverse of $\mathcal{H}(\hat{\theta})$
$\hat{\ell}^E(\theta; y_{1:I}^o)$	Ellipse approximation of the normalised log-likelihood function
$\hat{\ell}_p^E(\psi; y_{1:I}^o)$	Ellipse approximation of the normalised profile log-likelihood function

1 Introduction

Using mathematical models to understand data and the wider world is a fundamental tool in science and technology. Recently, a systematic workflow called *Profile-Wise Analysis* (PWA) has been put forward by Simpson and Maclaren [1] for identifiability analysis, parameter estimation and prediction of models given observed data. The workflow is presented as an alternative and complement to Bayesian workflows [6]. It provides a means to more efficiently perform identifiability analysis, which could be used in conjunction with a Bayesian workflow [7]. However, as an alternative to Bayesian workflows, its primary value is the ability to efficiently form frequentist confidence sets, which have approximate coverage guarantees for parameters and predictions, as opposed to credible intervals which have alternative aims in terms of representing beliefs [1, 8]. These prediction confidence sets also have approximate curvewise (simultaneous) coverage guarantees for model trajectories [1, 9], as opposed to alternative efficient frequentist prediction sets, which only have pointwise guarantees [10, 11]. It also provides value in allowing attribution of uncertainty in different parts of prediction space to specific parameters or combinations of parameters [1].

So far, the PWA workflow has been applied to mathematical biology models to assess parameter and model identifiability [1, 2, 9, 12–16], make predictions of the model trajectory/mean [1, 2, 9, 15, 16] and predictions of observations [9]. This includes deterministic models of the form ‘deterministic mathematical model + error model’ [1, 2, 9, 12, 15, 16] and stochastic models using an approximate auxiliary or surrogate likelihood [13, 14]. It also includes models where the data distribution depends solely on the mechanistic model through the distribution’s mean, like when modelling count data [17].

To demonstrate the statistical validity of the PWA workflow, in general, profile-wise prediction sets have been compared to full prediction sets qualitatively [1, 9, 16]. Additionally, the coverage performance of profile-wise confidence sets for the model trajectory has been tested in [9]. However, the computational performance of these sets, in conjunction with their statistical properties, has yet to be quantitatively assessed and remains an important contribution to demonstrate the value of the PWA workflow, which has been raised by reviewers of the PWA papers. Moreover, the introduction of tolerance sets for observations [18, 19] within the PWA workflow has been hinted at [1, 17] but remains to be formally introduced and robustly tested.

There is a software implementation of a version of the PWA workflow in *Julia* called `ProfileLikelihood` [20]. Here, we choose to create a separate implementation, `LikelihoodBasedProfileWiseAnalysis` [3], to explore more efficient heuristics for evaluating profiles, introduce tolerance sets for observations and provide implementations of functions for testing the statistical coverage

properties of the PWA workflow. This will allow the PWA workflow to be more accessible and easily tested against various models. The intention is not to replace other packages but to provide an outlet for the unique ideas and approaches seen throughout this thesis. The language *Julia* is chosen as it is a modern, flexible, high-performance language for numerical computing [21].

To follow along with the experiments in text see the relevant models in the examples section of `LikelihoodBasedProfileWiseAnalysis`' documentation [3].

1.1 Project Scope and Research Objectives

As outlined, the motivation of this thesis is to produce a software implementation of the PWA workflow as a *Julia* package that contains new features, improved methods for profile evaluation and functionality for performing tests of statistical coverage and computational performance. This package will then be used to robustly examine the statistical and computational properties of the PWA workflow.

Given this motivation, we define the following research objectives:

1. Provide a review of the PWA workflow.
2. Provide a review of tolerance intervals and their implementation within the PWA workflow.
Demonstrate the value of these intervals within the PWA workflow.
3. Make the PWA workflow more accessible by providing a software implementation as a *Julia* package with improved algorithms. Identify areas where these can be further improved, including using existing packages/methodologies (e.g. the CICO method [22] for bivariate boundaries or the InformationGeometry [23] approach in interest parameter space).
4. Apply the PWA workflow to existing (simple) examples in literature and give a more robust evaluation (quantitatively) of statistical properties and computational performance [in comparison to other methodologies].
5. Identify and evaluate methods to improve the statistical performance.
6. Apply the PWA workflow to non-trivial examples beyond those already discussed in literature.
 - (a) Higher-dimensional model with approximately ten parameters.
 - (b) Stochastic model (two-parameter birth and death model).

The last of these objectives is within scope; however, due to limited time and thesis length, we have restricted our focus to the first five to allow a more comprehensive consideration of these objectives. Instead, as part of the *Experiments* folder in `UoA_MastersWorking`, we have provided initial implementations of models for these goals, which can be explored in future work.

2 Literature Review

This literature review summarises the proposed PWA workflow from Simpson and Maclaren [1], which is further discussed in Murphy, Maclaren, and Simpson [9]. The PWA workflow looks to provide the ability to perform parameter identifiability, parameter estimation, prediction sensitivity with respect to parameters and construction of confidence and tolerance intervals for predictions. It also considers the background of the PWA workflow and other existing approaches in the literature. We build upon this work by shifting the focus of predictions of realisations to tolerance intervals rather than prediction intervals. This is implicitly considered in those works; here we consider it explicitly. Using these tolerance intervals as prediction intervals will still be considered. The software implementations of existing approaches will be discussed for each of the aspects the PWA workflow looks to solve.

Our discussion will centre around dynamic models in time, especially ODEs. In particular we focus on models of the form ‘deterministic mathematical model + error model’ as in Murphy, Maclaren, and Simpson [9]. The PWA workflow can be naturally extended to models where the data distribution depends on the mathematical model via the mean parameter only [17], as well as spatio-temporal [12] and stochastic models [13, 14]. Descriptions of the considered models using the notation outlined in Section 2.2 can be found in Appendix A.

2.1 Why Use The PWA Workflow: Types of Intervals and Sets

When conducting scientific research, it is important that the methods used have some degree of certainty regarding their accuracy. For example, imagine modelling the growth of a tumour. In particular, it may be of interest to estimate when (or even if) it would reach a certain critical mass, thus requiring an operation, given a set of initial observations of its size. A physician would need to have some degree of certainty in the method that is used in order to make a well-informed decision on when or if surgery would be required for this patient. This motivates using methods with properties that consistently enable a good understanding of the uncertainty in a given problem.

The PWA workflow seeks to solve this by providing intervals and sets with an associated confidence level [1]. A confidence level is a frequentist property, which means that when these sets are formed for repeated experiments, they are expected to contain the true values a particular proportion (confidence level) of the time [24]. For example, consider if our physician had to estimate the timing of when the tumour reached a certain critical mass for many patients, using a method that provides an interval for this timing with a 99% confidence level. Given this confidence level, we would expect the interval for the time to reach critical mass to contain the true time for approximately 99% of the assessed patients. Forming the interval at this confidence level thus provides a meaningful summary of information that

can be used to make well-informed decisions, such as when or if to perform surgery.

Within the PWA workflow, there are several types of intervals, as seen in Figure 1, each with slightly different properties and goals [25, 26]. In general, the corresponding set for each interval is all the values contained within the interval (inclusive). Our confidence sets are for parameters (both θ and the data distribution parameters ϕ). Our reference tolerance sets and prediction sets are instead for future observations. All of these sets, except the confidence set for θ , are sets for predictive quantities. The notation in this section will be explained in 2.2.

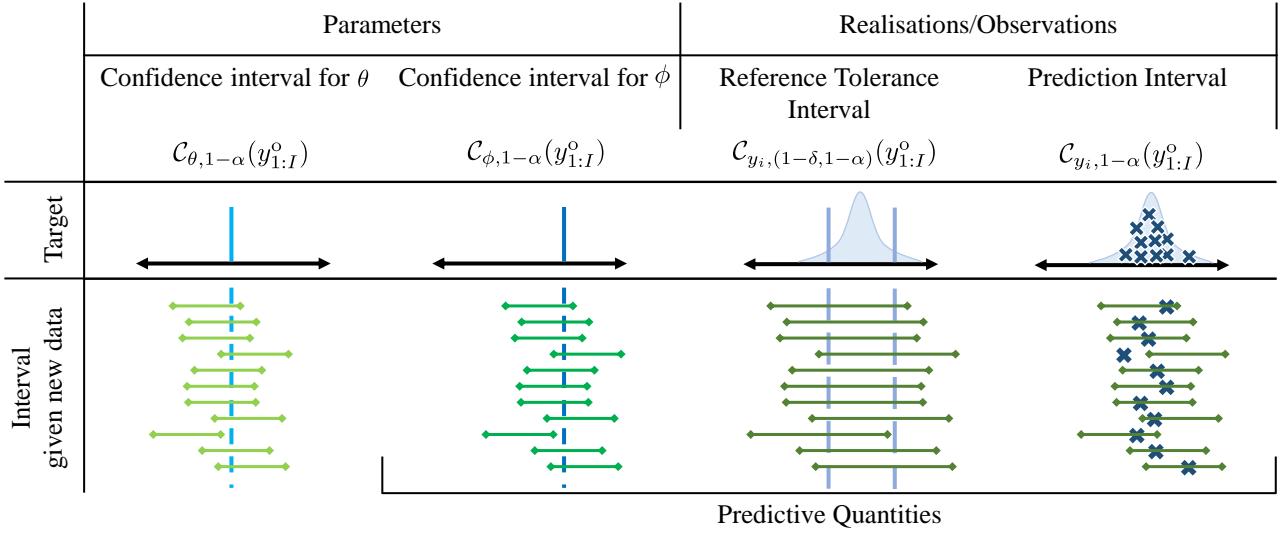


Figure 1: Types of intervals.

Figure 1 corresponds to the case where we apply the PWA workflow to a scalar model corrupted by Gaussian noise from a later section (Section 2.2.8.1), with minor modifications. Here we let $\theta^o = 1$ and let this value be known. We also slightly modify the scalar model so that $z(\theta^M) = 1.5\theta^M$. This means that the mapping onto data distribution parameters is given by $\phi(\theta) = (z(\theta^M), \theta^o) = (1.5 \times \theta^M, \theta^o)$.

2.1.1 For Parameters

We observe a set of observations from the model in Figure 1, which we then use to estimate the single parameter θ^M . The estimation of this single parameter uses a $1 - \alpha$ confidence interval, meaning that under repeated sampling of observations, $1 - \alpha$ of the intervals found using the estimation process are expected to contain the true value of θ^M [24]. Similarly, the confidence set for this single parameter includes all the values inside this interval (assuming the log-likelihood function is uni-modal).

We then propagate the confidence interval for θ into a confidence interval for ϕ using the mapping $\phi(\theta)$. Here, ϕ is an estimate of the underlying data distribution for observations. If we then wish to predict the true value of the model trajectory, $z(\theta)$, we consider the confidence interval for the first data distribution parameter (note, the second data distribution parameter is regarded as known). This is a

confidence interval for a predictive quantity; it has the same interpretation as the confidence interval for the parameter.

A practical example of where a parameter confidence interval could be of interest is modelling long-term coral coverage on a particular reef with little coverage after a disturbance [2]. In this example, one of the model parameters used is the long-time carrying capacity density. Forming a confidence interval for this parameter would thus provide an estimate of the uncertainty associated with this long-term coral coverage, e.g. it's likely to be between 75% and 83%.

Similarly, we might be interested in how the coverage, on average, increases towards the long-time carrying capacity. Our confidence set for $\phi(\theta)$ allows us to have a corresponding confidence set for the model trajectory, $z(\theta)$, which is taken to be the mean change in coral coverage over time. This then allows the estimation and visualisation of how long the average coverage will take to reach particular levels.

2.1.2 Predicting Observations

Now that we have a confidence interval for the only unknown data distribution parameter in Figure 1, we can use this interval in conjunction with the known parameter to make predictions for observations. The approach we introduce here is to form a $(1 - \delta, 1 - \alpha)$ reference tolerance interval for the $1 - \delta$ population reference interval. This means that under repeated sampling of observations, *at least* $1 - \alpha$ of the reference tolerance intervals we form using the prediction process are expected to contain the interval containing the $1 - \delta$ highest density region for new observations, as determined by the time point. As θ^o is known and set to its true value in this example, we would expect the coverage of the reference tolerance interval to be the same as the coverage of the confidence interval for the model trajectory.

If, instead, we wish to predict single future observations as a one-off task, then we can form prediction intervals [1]. In contrast to the reference tolerance interval, which seeks to trap the $1 - \delta$ population reference interval, a prediction interval seeks only to trap a single future observation at the desired coverage rate. This means that under repeated sampling of observations, *at least* $1 - \alpha$ of the prediction intervals formed are expected to contain a single new observation (where each set of observations is only used to predict one new observation). Resultantly, they are not necessarily as wide or required to be as wide as a reference tolerance interval [26].

Returning to the coral coverage example, the reference tolerance set describes the set of noisy observations we might take of the coral coverage over time. The considered reef area, as well as the cost of conducting these observations, is quite large [2], and hence, there is an associated level of variability

in a given set of observations. The range of observations that we most expect to see as time progresses is given by the $1 - \delta$ population reference set. Forming the reference tolerance set, thus, allows us to estimate the location of the population reference set with a specified level of confidence. We would expect $1 - \alpha$ of reference tolerance sets formed using repeated sets of observations to contain the most likely set of observations over time, the population reference set.

Similarly, if we wish to predict a single observation of the coral area coverage at some future time point, such as in one year, we would form a prediction interval with confidence level $1 - \alpha$. This means that if we were to go back many times and repeat our observations of coral coverage, which are used to form the prediction interval for the coverage in one year, we expect $1 - \alpha$ of these intervals to contain the observation made in one year.

2.2 PWA Workflow Formulation

In this section, we define the core components of the PWA workflow and notation. The notation will follow the same form as Simpson and Maclaren [1] and Murphy, Maclaren, and Simpson [9]. The background of the PWA workflow and potential implementations is discussed in Section 2.3. In this thesis we will primarily refer to the various sets and mathematical functions using their names rather than the symbols to help with readability.

2.2.1 Observed Data

Within the PWA workflow, as applied to mechanistic dynamic (ODE) models, observed data y_i^o is measured at discrete time points t_i . The ‘o’ superscript distinguishes the observed data from the random variable y that generated the data. Given $I \in \mathbb{N}$ observations, where $i = 1, 2, 3, \dots, I$, observed data is collected into the vector, $y_{1:I}^o \in \mathbb{R}^I$, which corresponds to the time points $t_{1:I}$. For multiple observations from the same time point and model component distinct indices in $1 : I$ are used. For example, if we have only two observations which are measured at $t = 1$ the first of these will correspond to $t_1 = 1$ and the second will correspond to $t_2 = 1$. For an observation of multiple model components at the same time point, we will use the same index in $1 : I$, such that, e.g. $y_i^o = (x_i^o, y_i^o)$. The same principles have been applied to spatio-temporal data as well [12].

2.2.2 Mechanistic Mathematical Model

The deterministic dynamic mechanistic models considered here are assumed to satisfy differential equations of the form:

$$\frac{dz}{dt} = f(z; \theta^M), \quad (1)$$

where θ^M is a vector of mechanistic model parameters, and z is a scalar or vector of model solutions containing no error. Model solutions evaluated at discrete time points t_i are defined as $z_i(\theta^M) = z(t_i; \theta^M)$. We will also refer to this as the model trajectory. In the same way as observations, $z_{1:I}(\theta^M)$ is a vector of the model solution evaluated at $t_{1:I}$, while $z(\theta^M)$ represents the continuous model solution.

2.2.3 Data Distribution Parameters

The PWA workflow assumes that the observable data distribution can be characterised by a data/statistical model with parameters $\phi(\theta)$. Here, ϕ is an ‘auxiliary’ function linking the mechanistic model and any other parameters to the data distribution parameters $\phi(\theta)$. The data distribution can be considered to come from an underlying mechanistic model with measurement error [1, 9, 12, 15], although representations for count data [12] and stochastic models [13, 14] can be considered as well. Examples are given below.

If we assume that the observed data come from a distribution that is dependent on the mechanistic model only through its trajectory, then we let:

$$\phi(\theta) = (z(\theta^M), \theta^o), \quad (2)$$

where typically (and here) the model solution is used as the mean or median parameter of the data distribution (error model) and θ^o is any additional observation parameters such as the observation error standard deviation. This gives us the full parameter vector, $\theta = (\theta^M, \theta^o)$, for the mechanistic model and error model. For simplicity, the additional observation parameters, θ^o , may be specified and regarded as known, as in Simpson and Maclaren [1]. However, unless they are truly known (as in, e.g. a synthetic coverage experiment), the likelihood function discussed in Section 2.2.4 is known as an estimated likelihood function because it does not account for the uncertainty in θ^o [24].

This then allows us to define data distributions evaluated at discrete time points t_i :

$$\phi_i(\theta) = (z_i(\theta^M), \theta^o), \quad (3)$$

where $z_i(\theta^M)$ is the mechanistic model solution at observation i .

In general, given data distribution parameters, we obtain a density function for the observed data y dependent on parameters, θ , of the form:

$$y \sim p(y; \theta) = p(y; \phi(\theta)). \quad (4)$$

When measured at time point t_i , this becomes:

$$y_i \sim p(y_i; \theta) = p(y_i; \phi_i(\theta)). \quad (5)$$

When the data distribution parameters are for an error model such as the additive Gaussian, log-normal and logit-normal models, the density function for y can be represented in the following ways.

For the additive Gaussian model, this is represented as [9]:

$$y_i \sim p(y_i; \theta) = \mathcal{N}(\phi_i(\theta)) = \mathcal{N}(z_i(\theta^M), \theta^o) = \mathcal{N}(z_i(\theta^M), \sigma_N^2), \quad (6)$$

where $\theta^o = \sigma_N$. This is equivalent to representing the observed data as the model trajectory plus error from a normal distribution with zero mean and variance, σ_N^2 .

For the log-normal model, this is represented as [9]:

$$y_i \sim \text{LogNormal}(\log(z_i(\theta^M)), \sigma_L^2), \quad (7)$$

where $\theta^o = \sigma_L$.

The logit-normal model is represented similarly to the log-normal model, noting that the mechanistic model solution is a proportion defined $\in (0, 1)$:

$$y_i \sim \text{LogitNormal}(\text{logit}(z_i(\theta^M)), \sigma_L^2), \quad (8)$$

where $\theta^o = \sigma_L$ and $\text{logit}(p) = \log(p/(1 - p))$.

2.2.4 Likelihood Function

If we have a vector of independent observations $y_{1:I}^o$ from our density function, y , which is a function of parameter θ , we can define the normalised likelihood function, $\hat{\mathcal{L}}(\theta; y_{1:I}^o)$, and in particular the normalised log-likelihood function, $\hat{\ell}(\theta; y_{1:I}^o)$. The ‘hat’, $\hat{\cdot}$, on \mathcal{L} and ℓ is used to represent that the functions are normalised. The likelihood function represents a measure of fit between the observed data and a particular parameter vector, which can be used to form confidence sets, e.g. for parameters [1, 24].

The normalised likelihood function is [27]:

$$\hat{\mathcal{L}}(\theta; y_{1:I}^o) = \frac{p(y_{1:I}^o; \phi(\theta))}{\sup_{\theta} p(y_{1:I}^o; \phi(\theta))}. \quad (9)$$

where the numerator is the likelihood function evaluated for data distribution parameters $\phi(\theta)$ at observations $y_{1:I}^o$ and the denominator is the maximum likelihood estimate (MLE) of the likelihood function as θ varies. The MLE will be estimated using numerical optimisation, with bounds on parameters, which have value $\hat{\theta}$ at the MLE.

Here, we will work with the normalised log-likelihood function of the form:

$$\begin{aligned}\hat{\ell}(\theta; y_{1:I}^o) &= \log \hat{\mathcal{L}}(\theta; y_{1:I}^o) \\ &= \log p(y_{1:I}^o; \phi(\theta)) - \sup_{\theta} \log p(y_{1:I}^o; \phi(\theta)) \\ &= \sum_{i=1}^I \log p(y_i^o; \phi(\theta)) - \sup_{\theta} \sum_{i=1}^I \log p(y_i^o; \phi(\theta)).\end{aligned}\tag{10}$$

Normalisation of the log-likelihood function means that $\hat{\ell}(\theta; y_{1:I}^o) \leq 0$ and $\hat{\ell}(\hat{\theta}; y_{1:I}^o) = 0$.

As in Murphy, Maclaren, and Simpson [9], we will use the `loglikelihood` function implemented in the *Julia Distributions* package [28] to determine the value of the log-likelihood function. This is straightforward to compute for the error models specified in Section 2.2.3 given observed data $y_{1:I}^o$. We expect that this function will be continuous for the models considered.

2.2.5 Profile Likelihood Function

Given a likelihood function, we can define a profile log-likelihood function as a function of so-called interest parameters, ψ , with nuisance parameters, ω , ‘eliminated’ [1]. These parameters represent a partitioning of the parameter vector $\theta = (\psi, \omega)$. The normalised profile log-likelihood function for ψ is then:

$$\hat{\ell}_p(\psi; y_{1:I}^o) = \sup_{\omega \mid \psi} \hat{\ell}_p(\psi, \omega; y_{1:I}^o),\tag{11}$$

where for each value of the interest parameter ψ , the values of ω are optimised out, meaning they are set to the values, $\omega^*(\psi)$, that maximise the function for each ψ . This defines a curve/hyper-surface which traces a ‘profile path’ in parameter space [1]. Resultantly, the values of ω vary with ψ along the hyper-surface, rather than being set to their global MLE values. We expect that this function will be continuous for the models considered.

2.2.6 Confidence Sets For Parameters

Using the log-likelihood function, we can define approximate likelihood-based confidence sets for the full parameter vector θ [24, 27]:

$$\mathcal{C}_{\theta, 1-\alpha}(y_{1:I}^o) = \{\theta \mid \hat{\ell}(\theta; y_{1:I}^o) \geq \ell_c\},\tag{12}$$

where ℓ_c is a threshold chosen so that the approximate coverage of the confidence interval is $1 - \alpha$. This coverage means that when repeating the experiment many times, we would expect $1 - \alpha$ of these confidence intervals to contain the true parameter [24]. For sufficiently regular problems (see Section 2.3.1) the threshold is calibrated using the chi-square distribution [24]:

$$\ell_c = -\frac{\Delta_{\nu, 1-\alpha}}{2},\tag{13}$$

where $\Delta_{\nu,1-\alpha}$ is the $1 - \alpha$ quantile of the χ^2 distribution with ν degrees of freedom. For brevity, we will refer to these as full parameter confidence sets. For full parameter confidence sets, we set ν equal to the number of parameters, $|\theta|$.

Profile likelihood-based confidence sets for the interest parameter(s) ψ take the form:

$$\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \{\theta = (\psi, \omega) \mid \hat{\ell}_p(\psi; y_{1:I}^o) \geq \ell_c\}, \quad (14)$$

where we instead set ν equal to the dimensionality of the interest parameters (e.g. for confidence sets with a single interest parameter $\nu = 1$ and with two interest parameters $\nu = 2$). We also record the optimised out values of nuisance parameters, $\omega^*(\psi)$, at each ψ in this confidence set for each value of the interest parameter. Recording these nuisance parameters allows this set to be propagated forward into predictive quantities; they do not have to be recorded otherwise. Simultaneous profile likelihood-based confidence sets for interest parameters can be obtained by setting $\nu = |\theta|$ [29]. The simultaneous sets represent more conservative sets which have extremities that touch the extremities of the full parameter confidence sets.

For brevity, we will refer to profile likelihood-based confidence sets for one and two interest parameters as univariate profiles and bivariate profiles, respectively. If these profiles are instead created using the simultaneous (full likelihood) asymptotic threshold with $\nu = |\theta|$, they will be referred to as ‘simultaneous univariate’ or ‘simultaneous bivariate’ profiles.

2.2.7 Confidence Sets For Data Distribution Parameters

By propagating forward full parameter confidence sets and profile likelihood-based confidence sets using the mapping $\phi(\theta)$, we define approximate likelihood-based confidence sets for the data distribution parameters ϕ . We refer to the data distribution parameters, ϕ , as ‘predictive’ quantities [1]. We use square brackets in the following equations to show that the confidence set for data distribution parameters is the set image of the parameter confidence set under the mapping $\phi(\theta)$.

For the likelihood-based confidence set for data distribution parameters from full parameter confidence sets, this is defined as:

$$\mathcal{C}_{\phi,1-\alpha}(y_{1:I}^o) = \{\phi[\mathcal{C}_{\theta,1-\alpha}(y_{1:I}^o)]\} = \{\phi(\theta) \mid \theta \in \mathcal{C}_{\theta,1-\alpha}(y_{1:I}^o)\}. \quad (15)$$

This set definition implies that the confidence set $\mathcal{C}_{\phi,1-\alpha}(y_{1:I}^o)$ has at least a coverage of $1 - \alpha$ (is conservative), assuming $\mathcal{C}_{\theta,1-\alpha}(y_{1:I}^o)$ has $1 - \alpha$ coverage, given the following relationship [1]:

$$\theta \in \mathcal{C}_{\theta,1-\alpha} \implies \phi(\theta) \in \mathcal{C}_{\phi,1-\alpha}. \quad (16)$$

We define a profile likelihood-based confidence set, called a profile-wise confidence set, as:

$$\mathcal{C}_{\phi,1-\alpha}^{\psi}(y_{1:I}^o) = \{\phi[\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)]\} = \{\phi(\psi, \omega^*(\psi)) \mid (\psi, \omega^*(\psi)) \in \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)\}. \quad (17)$$

The profile-wise confidence set provides a confidence set for $\phi(\theta)$, assuming that uncertainty is primarily driven by ψ [1].

More conservative profile-wise confidence sets for data distribution parameters can be formed by taking the union of individual profile confidence sets:

$$\mathcal{C}_{\phi,1-\alpha} \approx \bigcup_{\psi} \mathcal{C}_{\phi,1-\alpha}^{\psi}. \quad (18)$$

This can be considered as a more conservative approximation to the confidence set based on the full likelihood. Additionally, increasing the dimension of the interest parameter will obtain a better approximation of this confidence set [1], because these higher dimensional profiles better capture the interactions between multiple parameters that are featured in the full parameter confidence set.

When considered in a mechanistic model context, this allows us to form a trajectory confidence set for the entire mechanistic model solution, which is typically treated as the mean or median data distribution parameter [1, 9]. If we wish to form a trajectory confidence set for the mechanistic model solution, we consider just that component of the data distribution confidence set. In this case, if the parameter confidence set has the correct coverage properties over all parameters simultaneously, we expect the trajectory confidence set, $\mathcal{C}_{\phi,1-\alpha}$, to display *curvewise* (simultaneous) coverage properties, where the true model solution is fully contained within the confidence set [9].

For brevity, we will refer to likelihood-based confidence sets for the model trajectory from full parameter confidence sets as full trajectory confidence sets. Similarly, we will refer to the sets for the data distribution parameters as full data distribution confidence sets. Additionally, we will refer to profile-wise confidence sets for the model trajectory from univariate and bivariate profiles as profile-wise trajectory confidence sets. We will specify which profiles the profile-wise trajectory confidence sets were formed using.

2.2.8 Reference Tolerance Sets For Observed Data

Building upon Simpson and Maclaren [1] and Simpson, Murphy, and Maclaren [17] we define $(1 - \delta, 1 - \alpha)$ reference tolerance sets for $1 - \delta$ population reference sets given observed data [18, 19]. The background of this approach will be further covered in Section 2.3.6.2.

A $1 - \delta$ population reference interval refers to an interval that contains $1 - \delta$ of the population (i.e. of observations) [18, 19]. These population reference intervals are also ‘predictive’ quantities. Here,

a $1 - \delta$ population reference set refers to the set of $1 - \delta$ population reference intervals across time points, t_j (and similarly for reference tolerance sets and reference tolerance intervals). Such reference sets are not unique and so, in general, we will take a $1 - \delta$ population reference interval to be given by the $1 - \delta$ highest density region [30] of population observations at t_j . This will allow the interval to represent a ‘typical’ observation best.

Similarly, a $(1 - \delta, 1 - \alpha)$ reference tolerance interval is a tolerance interval that contains the $1 - \delta$ reference interval with probability $1 - \alpha$. We will refer to these as reference tolerance intervals unless the $(1 - \delta, 1 - \alpha)$ designation is important for clarity. These intervals can be used as approximate prediction intervals; they are often conservative for this task but they are not guaranteed to be valid prediction sets. In these cases, a Bonferroni correction can be used to get more conservative predictions for observations [1, 9, 31]. As our notation suggests, a reference tolerance interval can be formed that has a coverage property at a different confidence level to the size of the reference interval (i.e. $\delta \neq \alpha$). In this work, we will let $\delta = \alpha$.

Therefore, given a desired confidence level $1 - \alpha$, we form a $1 - \alpha$ likelihood-based confidence set for data distribution parameters, $\mathcal{C}_{\phi,1-\alpha}$, from the full parameter confidence set. Then, for each ϕ in the data distribution parameter set we construct a $1 - \delta$ reference set, $\mathcal{A}_{y,1-\delta}^\phi(y_{1:I}^o)$, where ϕ is related to y as in Equation (4). This reference set is constructed by taking a $1 - \delta$ region of the data distribution. For symmetric data distributions, we take the $\delta/2$ and $1 - \delta/2$ quantiles of the probability distribution. For asymmetric distributions, we take the $1 - \delta$ highest density region [30]. If the asymmetric distribution is unimodal, we can use the author’s *Julia* package `UnivariateUnimodalHighestDensityRegion` [5] to evaluate the highest density region. We then take the union across the reference sets formed from each ϕ to obtain $(1 - \delta, 1 - \alpha)$ reference tolerance sets for observed data, y^o , from full parameter confidence sets:

$$\mathcal{C}_{y,(1-\delta,1-\alpha)}(y_{1:I}^o) \approx \bigcup_{\phi \in \mathcal{C}_{\phi,1-\alpha}(y_{1:I}^o)} \mathcal{A}_{y,1-\delta}^\phi(y_{1:I}^o). \quad (19)$$

Because each $\mathcal{A}_{y,1-\delta}^\phi(y_{1:I}^o)$ can only be guaranteed to be equal to the population reference set if it was derived using the true parameter values, we refer to these only as *reference sets*. Similarly, because $\mathcal{C}_{y,(1-\delta,1-\alpha)}(y_{1:I}^o)$ is obtained by taking the union over each reference set, it is effectively a confidence set for the population reference set, i.e. a tolerance interval. Hence, we refer to these as *reference tolerance sets*. Usefully, if the data distribution parameter confidence set, $\mathcal{C}_{\phi,1-\alpha}$, has curvewise coverage properties then we also expect the reference tolerance set, $\mathcal{C}_{y,(1-\delta,1-\alpha)}(y_{1:I}^o)$, to have curvewise coverage properties in the sense of trapping the full curvewise population reference set. We will refer to these as full reference tolerance sets.

We do the same thing for profile-wise reference tolerance sets, beginning from profile-wise data distribution confidence sets:

$$\mathcal{C}_{y,(1-\delta,1-\alpha)}^{\psi}(y_{1:I}^o) \approx \bigcup_{\phi(\psi,\omega^*(\psi)) \in \mathcal{C}_{\phi,1-\alpha}^{\psi}(y_{1:I}^o)} \mathcal{A}_{1-\delta}^{\phi(\psi,\omega^*(\psi))}(y_{1:I}^o). \quad (20)$$

More conservative profile-wise reference tolerance sets can again be formed by taking the union of reference tolerance sets from individual profiles [1]:

$$\mathcal{C}_{y,(1-\delta,1-\alpha)} \approx \bigcup_{\psi} \mathcal{C}_{y,(1-\delta,1-\alpha)}^{\psi}. \quad (21)$$

We will refer to these as profile-wise reference tolerance sets and specify which profiles the sets were formed using.

2.2.8.1 Full Reference Tolerance Set Coverage

The coverage of the full reference tolerance set in the PWA workflow is straightforward to prove. The $(1 - \delta, 1 - \alpha)$ reference tolerance set is constructed from the $1 - \alpha$ confidence set for data distribution parameters, which is constructed from the $1 - \alpha$ confidence set for parameters. Here let $1 - \delta = 1 - \alpha = 0.95 = 95\%$. If a full parameter confidence set has 95% coverage and the model is well-specified, then 95% of the time it will trap the true parameters. Therefore, the full data distribution confidence set will also have *at least* 95% coverage (Equation (16)). Additionally, the true data distribution parameters give the true $1 - \delta$ reference interval for the population at each time point. Hence, if the full data distribution confidence set has *at least* 95% coverage, then our full reference tolerance set will contain the 95% reference set *at least* 95% of the time.

In the mechanistic model case, if the full data distribution confidence set has curvewise coverage of the model solution, then our full reference tolerance set will also have curvewise coverage of the population reference set. That is, 95% of the full reference tolerance sets constructed in this fashion under repeated sampling of new data will contain the 95% population reference set (all 95% population reference intervals). A similar idea to what we demonstrate here, but in a different context, is found in Satten [32].

For example, consider a single parameter scalar model, $z(\theta^M) = \theta^M$, from which we have obtained $I = 100$ observations corrupted by i.i.d. Gaussian noise $\sim \mathcal{N}(0, \theta^o)$. Let $\theta^M = 2$ and $\theta^o = 1$. The density function for observations is represented as:

$$y \sim p(y; \theta) = \mathcal{N}(\phi(\theta)) = \mathcal{N}(z(\theta^M), \theta^o) = \mathcal{N}(z(\theta^M), \sigma), \quad (22)$$

where $\theta^o = \sigma$ and $\theta = (\theta^M, \theta^o)$. Resultantly, our parameter vector, θ , has two parameters.

The 95% population reference interval can be formed by considering the 2.5% and 97.5% quantiles of this density function under the true parameterisation, $\theta = [0.040, 3.960]$. Then, given the I observations we form a 95% confidence set for θ , $\mathcal{C}_{\theta,0.95}$ (Figure 2a). The mapping of θ onto the data distribution parameters ϕ is 1:1, hence we also have a 95% confidence set for ϕ , $\mathcal{C}_{\phi,0.95}$. We then use each $\phi \in \mathcal{C}_{\phi,0.95}$ to form 95% reference intervals, $\mathcal{A}_{y,0.95}^\phi$ (Figure 2b). We then take the union across each of these reference intervals to form (95%, 95%) reference tolerance intervals, $\mathcal{C}_{y,(0.95,0.95)}$ (Figure 2c). If the true parameterisation is in $\mathcal{C}_{\theta,0.95}$ with 95% coverage, then it is also in $\mathcal{C}_{\phi,0.95}$ with 95% coverage and our full reference tolerance interval, $\mathcal{C}_{y,(0.95,0.95)}$, will also contain the *at least* 95% reference interval with *at least* 95% coverage.

In our example in Figure 2a, the true parameter vector is in our full parameter confidence set, so our full reference tolerance interval in Figure 2c encloses the population reference interval. The same example, but using univariate profiles, can be seen in Appendix B.

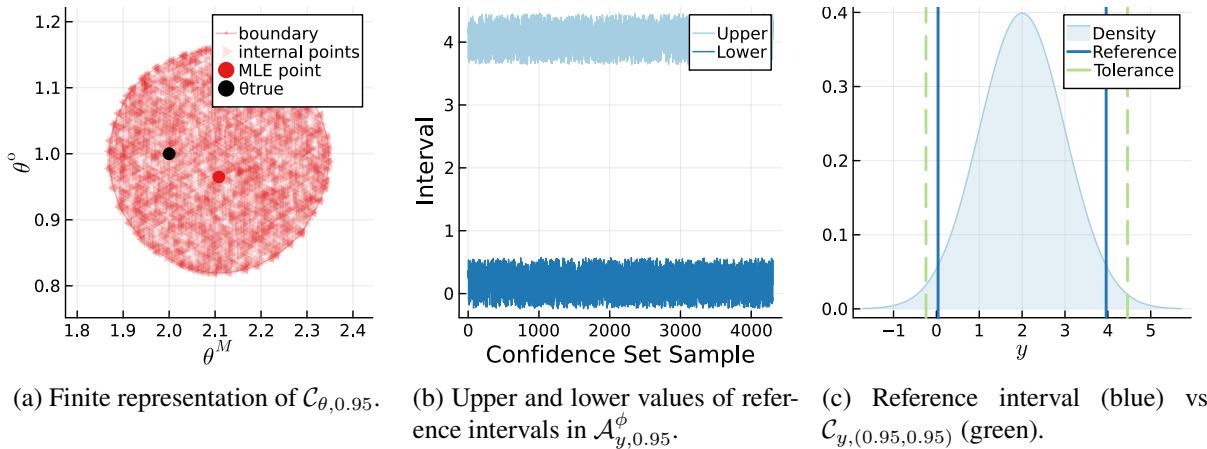


Figure 2: PWA workflow applied to obtain reference tolerance interval $\mathcal{C}_{y,(0.95,0.95)}$, which encloses the 95% population reference interval, given 100 observations of y .

The ‘at least’ statement for both the reference interval and the coverage of this interval occurs for the following reasons. For the coverage statement, if $\theta \notin \mathcal{C}_{\theta,0.95}$, then it is possible that we predict the population reference interval using the incorrect parameter or similarly from the union of reference intervals from many incorrect parameters. Similarly, for the reference interval statement, if $\theta \in \mathcal{C}_{\theta,0.95}$, then our 95% reference interval from that θ is the 95% population reference interval. Other parameter values in the parameter confidence set may predict reference intervals outside of this range. Hence, the union of these intervals will contain at least the 95% population reference interval.

2.2.9 Numerical Implementation

In order to assist optimisation algorithms and specify parameter regions to evaluate points from within, we define the following bounds on parameters, $\theta \in [a, b]$. We can also specify nuisance

parameter bounds that are different from $[a, b]$ (i.e. contained within $[a, b]$, but with a smaller range) to assist with numerical optimisation, but in general, we will keep them the same. We use the *Julia Optimization* package [33] for numerical optimisation of nuisance parameters of profile log-likelihood functions. As in Simpson and Maclaren [1] and Murphy, Maclaren, and Simpson [9], we use the MLE value of nuisance parameters as their initial guess when conducting numerical optimisation. The *Optimization* package can use optimisation algorithms from many different backends and can specify automatic differentiation backends for algorithms that use derivatives. Within *Optimization*, we use the *NLopt* optimisation package [34] and the *BOBYQA* algorithm [35].

2.3 PWA Workflow Background and Existing Methods

The PWA workflow is a Fisherian approach to statistical inference, which provides a form of Bayesian-frequentist compromise [24]. It provides intervals and sets with the desired frequentist coverage properties through the likelihood function. This means that these sets when formed for repeated experiments are expected to contain the true values a particular proportion (confidence level) of the time [24]. The approach can be connected with the Bayesian approach if the priors/bounds for model parameters θ are treated as uniform priors: then the likelihood function encodes the information on θ from observing data $y_{1:T}^0$. If we then scale the likelihood function to integrate to one, the likelihood function and posterior density function are equivalent [24].

Bayesian workflows can be used in conjunction or as an alternative to the proposed PWA workflow. Parameter identifiability [6, 7, 12, 36] and prediction [6] can be performed using Bayesian workflows, making use of Markov Chain or Hamiltonian Monte Carlo sampling methods. However, profile likelihood methods may be more appropriate than Bayesian methods for assessing parameter identifiability and determining how to constrain parameters to allow efficient Monte Carlo sampling [7, 12]. Resultantly, using the profile likelihood approach for identifiability analysis before switching to the Bayesian approach for the remainder of the analysis can be a good strategy.

2.3.1 Ellipse/Elliptical Approximation of the Log-Likelihood Function

A common approach for approximating the log-likelihood function to enable faster inference and generation of approximate parameter confidence intervals is the use of the Fisher matrix (FM) [37, 38]. This is also known as the Fisher information matrix (FIM) [24]. It has two slightly different definitions: the observed and expected FIM [24]. Here, we are interested in the observed FIM. The observed FIM is a quadratic approximation of the curvature of the log-likelihood function at the MLE. Here, we will represent it using the notation $\mathcal{H}(\hat{\theta})$, as I is already defined. In the single parameter case, the observed Fisher information is the negative of the second derivative of the log-likelihood

function evaluated at the MLE [24]:

$$\mathcal{H}(\hat{\theta}) \equiv -\frac{\partial^2}{\partial \theta^2} \ell(\hat{\theta}; y_{1:I}^o). \quad (23)$$

In the multi-parameter case, it is the matrix of second derivatives (the Hessian) of the log-likelihood function evaluated at the MLE with elements [24]:

$$H_{jk}(\hat{\theta}) \equiv -\frac{\partial^2}{\partial \theta_j \partial \theta_k} \ell(\hat{\theta}; y_{1:I}^o). \quad (24)$$

This then allows us to define the following approximation of the normalised log-likelihood function using a second-order Taylor expansion at the MLE [24]:

$$\begin{aligned} \hat{\ell}(\theta; y_{1:I}^o) &\approx \hat{\ell}^E(\theta; y_{1:I}^o) \\ &\approx -\frac{1}{2}(\theta - \hat{\theta})' \mathcal{H}(\hat{\theta})(\theta - \hat{\theta}), \end{aligned} \quad (25)$$

where the other terms in the Taylor expansion are consumed by the normalisation.

$\mathcal{H}(\hat{\theta})$ can be evaluated numerically using finite differencing or more accurate automatic differentiation methods [39, 40]. Here, we will use the `ForwardDiff Julia` package [41], which implements forward-mode automatic differentiation.

This approximation will be reasonable if the likelihood function is sufficiently regular, i.e., it can be well approximated by a quadratic function around the MLE [24]. Additionally, for large sample sizes, the likelihood function will generally become more regular [24].

If we set one side of Equation (25) equal to 1 at some target log-likelihood value, ℓ_c , then it is directly equivalent to the equation of a $|\theta|$ dimensional ellipsoid [42] (variable names changed for consistency):

$$-\frac{1}{2\ell_c}(\theta - \hat{\theta})' \mathcal{H}(\hat{\theta})(\theta - \hat{\theta}) = 1. \quad (26)$$

This result is particularly useful in two dimensions, as it allows us to derive an exact equation and solution for finding points on the ellipsoid at target value $\hat{\ell}_c$. Resultantly, here and within `LikelihoodBasedProfileWiseAnalysis`, we will refer to the observed Fisher information matrix as the ellipse approximation of the log-likelihood function at the MLE.

2.3.1.1 Profile Likelihood Approximation

The same approximation can be used to approximate the profile log-likelihood function. In order to eliminate nuisance parameters, we must first invert the Hessian matrix to obtain the covariance matrix $\Gamma(\hat{\theta}) = H^{-1}(\hat{\theta})$. Then, for parameter j the curvature of the profile log-likelihood is $\Gamma_{jj}(\hat{\theta})^{-1}$ [24]. In general, $\Gamma_{jj}(\hat{\theta})^{-1} < H_{jj}(\hat{\theta})$ as the former takes into account that the nuisance parameters are unknown and the latter assumes the nuisance parameters are known at their MLE value [24].

If some of the parameters of the log-likelihood function are non-identifiable (see Section 2.3.2), then

$\mathcal{H}(\hat{\theta})$ will be singular and not invertible [43]. Because we evaluate the Hessian matrix using numerical methods, it may not be singular when it should be. Therefore, we can test for singularity by evaluating the rank of $\mathcal{H}(\hat{\theta})$ or checking if its standardised eigenvalues are close to zero, where standardised eigenvalues are the matrix eigenvalues divided by the largest eigenvalue [43]. In the singular case, the profile likelihood approximation does not exist. When this occurs, we will use a pseudo-inverse (as in [44, 45]) to invert the Hessian to avoid functions breaking in `LikelihoodBasedProfileWiseAnalysis`, but warn that outputs are unlikely to be correct.

If the Hessian is invertible then univariate parameter profiles with $\psi = \theta_j$ have the quadratic approximation [24]:

$$\begin{aligned}\hat{\ell}_p(\psi; y_{1:I}^o) &\approx \hat{\ell}_p^{\mathcal{E}}(\psi; y_{1:I}^o) \\ &\approx -\frac{1}{2}\Gamma_{jj}(\hat{\theta})^{-1}(\psi - \hat{\psi})^2.\end{aligned}\tag{27}$$

Similarly, if we are interested in only a portion of parameter space, such as a bivariate profile with parameters $\theta_j \cup \theta_k = \psi$, then the quadratic approximation will be:

$$\begin{aligned}\hat{\ell}_p(\psi; y_{1:I}^o) &\approx \hat{\ell}_p^{\mathcal{E}}(\psi; y_{1:I}^o) \\ &\approx -\frac{1}{2}(\psi - \hat{\psi})'([e_j, e_k]' \Gamma(\hat{\theta}) [e_j, e_k])^{-1}(\psi - \hat{\psi}), \quad \theta_j \cup \theta_k = \psi,\end{aligned}\tag{28}$$

where e_j and e_k are the j th and k th canonical vectors of $\mathbb{R}^{|\theta|}$. Then $([e_j, e_k]' \Gamma(\hat{\theta}) [e_j, e_k])^{-1}$ is the inverse of the two by two matrix from row and column j and k of $\Gamma(\hat{\theta})$.

These equations are equivalent to the case where we calculate the profile likelihood using the full Hessian matrix $\mathcal{H}(\hat{\theta})$ in Equation (25) with the following requirements. Again, the functions are continuous. We require that nuisance parameters are set to their optimal value for a given interest parameter value, there are no bounds on parameters, and there are no issues with the invertibility of $\mathcal{H}(\hat{\theta})$. This is shown in Section 3.1.2.1 for univariate profiles.

2.3.1.2 Improved Approximations

Improved approximations that use higher order Taylor expansions of the likelihood function around $\hat{\theta}$ exist, such as the DALI method (Derivative Approximation for Likelihoods) [37, 46, 47]. In cases where the likelihood function is not well-represented by the ellipse approximation (is non-Gaussian in the parameter space), the DALI method better captures the non-Gaussianity.

2.3.2 Parameter Identifiability and Sloppiness

It is often of interest to use data to estimate the values of parameters that allow a model to match the data well. This leads to a discussion of parameter identifiability, which seeks to answer whether parameters can be well determined given experimental data. A parameter can either be structurally

non-identifiable, practically non-identifiable or identifiable. These different identifiability cases when considering the profile log-likelihood function can be seen in Figure 3. Additionally, parameter identifiability can be important for enabling the feasibility of other analysis methods, such as Bayesian Markov Chain Monte Carlo sampling for assessing uncertainty in parameters and model predictions [7] and making accurate predictions of some model variables [48, 49]. Identifiability is also connected with the idea of sloppiness [43, 50]. However, it is not equivalent; sloppiness refers to parameters which have little impact on the model output while varying by orders of magnitude [50].

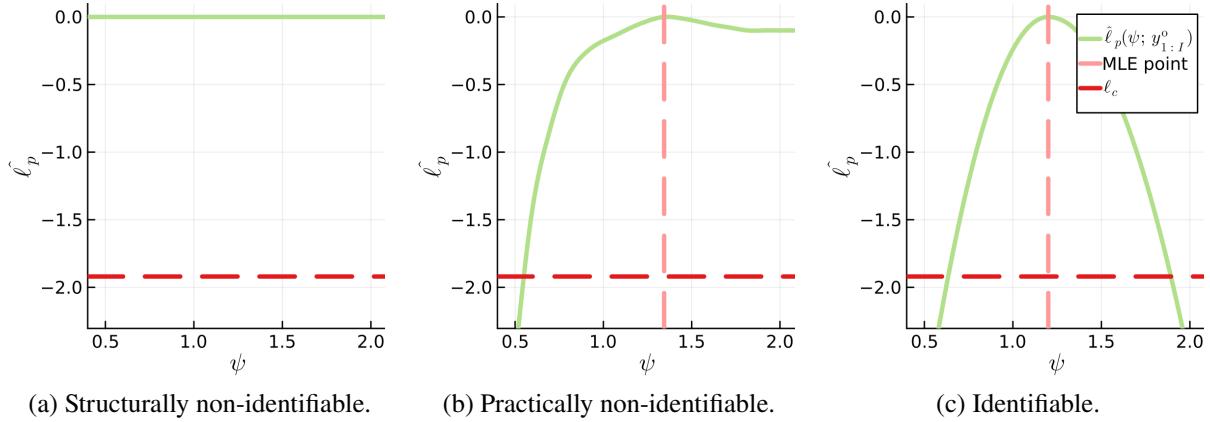


Figure 3: Identifiability cases for the normalised profile log-likelihood function (solid green) of some interest parameter ψ . The vertical line (dotted light red) is the MLE value. The horizontal line (dotted dark red) is the value of ℓ_c that corresponds to 95% likelihood-based confidence intervals (intersection with the green line).

2.3.2.1 Structurally Non-Identifiable Parameters

A parameter will be structurally non-identifiable if, under the experimental design, the observed data can be explained equally well by many different values of the same parameter [29, 48]. An alternative but equivalent definition is that a parameter is structurally non-identifiable if it cannot be uniquely determined from infinite error-free observations [51]. In a univariate profile likelihood approach (one parameter), this is identified as a flat line across parameter values (Figure 3a). An example of this is estimating the values of the parameters in the equation $y = X_1 + X_2$ for an observed model solution of $y = 5$. Without any additional information on the value of either X_1 or X_2 , or how they are related, both parameters are structurally non-identifiable. This is because any value of each parameter can fit the observed model solution equally well so long as the other parameter follows the relationship of $X_1 = 5 - X_2$. However, a model reparameterisation using parameter X_3 , where $X_3 = X_1 + X_2$, is identifiable. Some examples of structural non-identifiability in ODE models are given by Raue et al. [7] and Fröhlich, Theis, and Hasenauer [52].

Identifying the structural non-identifiability of parameters in deterministic models where data is not

affected by observation noise or error is a relatively well-defined problem for ODE models where many software solutions are available [49]. However, it is less well explored for PDE models [9] and other models. Software for ODE models include the *Julia* packages `StructuralIdentifiability` [53] and `SIAN` [54, 55] and the *MATLAB* toolbox `STRIKE-GOLDD` [56]. `StructuralIdentifiability` also can assess the identifiability of functions of parameters, using some of the algorithms from Ovchinnikov et al. [57]. Similarly, `STRIKE-GOLDD` can also identify good reparameterisations for non-identifiable parameters.

2.3.2.2 Practically Non-Identifiable Parameters

A parameter will be practically non-identifiable if the profile log-likelihood function has a unique maximum yet has a likelihood-based confidence interval that is uselessly large at a confidence level of interest [29]. Practical non-identifiability indicates an insufficient amount of or quality of experimental data, which is not error-free. In a univariate profile likelihood approach, this is seen as a profile with a unique peak that decreases before plateauing on one or both sides of the peak above a desired confidence level threshold, ℓ_c [7, 48] (Figure 3b). This can mean that a parameter is practically identifiable up to a certain confidence level but is practically non-identifiable beyond that level [15, 29]. Some examples of practical non-identifiability in models are given by Simpson et al. [12], Raue et al. [29], and Simpson et al. [58].

Identifying the practical non-identifiability of parameters in models where data is affected by observation error has many software solutions that use variations of the profile likelihood approach, although other approaches exist. Software solutions that use the profile likelihood approach include the *MATLAB Data2Dynamics* toolbox [59] and the *Julia* packages `ProfileLikelihood` [20] and `LikelihoodProfiler` [22]. Other approaches include the Identifiability-Test by Radial Penalization [60] and Bayesian approaches for nonlinear biophysical models and disease transmission modelling [6, 36].

2.3.2.3 Identifiable Parameters

A parameter will be identifiable if it is not structurally or practically non-identifiable, meaning it will have a usefully finite likelihood-based confidence interval [29] (Figure 3c). If a parameter is non-identifiable, identifiability analysis can be used to specify how an experimental design can be updated to resolve structural and practical identifiability issues [7, 29].

2.3.2.4 Model Reduction

Identifiability analysis can be used to inform appropriate model reductions, which can lead to improved computational performance [61]. For example, a set of structurally non-identifiable parameters can

be replaced with a combined ‘lumped’ parameter that is identifiable [43, 62]. Similarly, in FIM-based methods, it may be desirable to fix non-identifiable parameters to some reasonable value in order to make $\mathcal{H}(\hat{\theta})$ invertible [44]. This is similar to sensitivity analysis approaches [62]. Other methods for model reduction include timescale methods and singular value decomposition (SVD) [62], of which principal component analysis (PCA) is a special case [50].

2.3.3 Parameter Confidence Intervals

Confidence intervals for parameters can be estimated using the profile log-likelihood function or the ellipse approximation of that function [24, 29]. They can also be estimated using bootstrap methods [63], although these may perform poorly for models with practically non-identifiable parameters [52].

In order to evaluate these intervals, we define a confidence threshold ℓ_c . This confidence threshold is calibrated using a chi-square distribution [10, 24] as stated in Section 2.2.6, which is also known as Wilk’s likelihood ratio statistic [10]. This calibration is reasonable if the problem is sufficiently regular or there is a sufficiently large number of data points such that the likelihood function becomes regular [64]. These intervals can be seen in Section 3.1.2.

2.3.3.1 Wald Confidence Intervals

Wald confidence intervals for parameters are calculated using the ellipse approximation of the profile log-likelihood function [24]. They are also known as asymptotic confidence intervals [29]. However, we will refer to these only as Wald confidence intervals so there is no misunderstanding with likelihood-based confidence intervals. For interest parameter θ_j at confidence threshold ℓ_c these are given by [29]:

$$\sigma_j^\pm = \hat{\theta}_j \pm \sqrt{-2\ell_c \times \Gamma_{jj}(\hat{\theta})}, \quad (29)$$

where σ_j^\pm is the Wald confidence interval. Note that $-2\ell_c$ is positive due to the definition of ℓ_c in Equation (13).

If the problem is sufficiently regular, these intervals will reasonably quantify the uncertainty in $\hat{\theta}$. If observations depend linearly on parameters, then they are exact [29]. However, if the parameter estimate is close to a boundary, the data are sparse, or the Hessian matrix is not invertible (see Section 2.3.1.1), these intervals will not be suitable [24]. Additionally, they may not be suitable for non-linear models and models with identifiability issues [29, 52, 63].

In cases where the log-likelihood function is not very regular, a parameter transformation, $\Theta = g(\theta)$, can be used to improve regularity and thus make Wald confidence intervals more reliable [24]. However, this does require knowledge of the regularising transformation.

2.3.3.2 Likelihood-Based Confidence Intervals

Likelihood-based confidence intervals use the same confidence threshold as Wald intervals but apply it to the profile log-likelihood function instead. These confidence intervals are the boundaries of the univariate profiles in Section 2.2.6 [24, 29]. These intervals are better than Wald intervals because they automatically use the best possible regularising transformation; it does not need to be known [24]. If an exact regularising transformation does not exist, the intervals will have the coverage of the best transformation. Hence, the exact coverage of these intervals is not guaranteed [24].

The boundaries of univariate profiles can be calculated by solving for each value of ψ that satisfies:

$$0 = \hat{\ell}_p(\psi; y_{1:I}^o) - \ell_c. \quad (30)$$

If there is not a solution to Equation (30) for $\theta_j = \psi$ available on each side of $\hat{\theta}_i$, then the parameter is practically non-identifiable at the confidence level of ℓ_c , as stated in Section 2.3.2.2. Similarly, if the solution for ψ is not available within the parameter bounds ($\psi \notin [a_j, b_j]$), the parameter will be practically non-identifiable at this confidence level within the bounds [22]. In the case of multi-modal univariate profiles, there can be multiple solutions on each side of $\hat{\theta}_j$ (e.g. similar to that seen for marginal posterior distributions of samples in [65]). For this thesis, we will assume that univariate profiles are unimodal (when identifiable), which is the case in the considered models.

2.3.3.3 Numerical Implementations

Numerical methods proposed and available for calculating confidence intervals fall into three main categories: step-wise, integration-based and constrained optimisation.

Step-wise approaches begin at $\hat{\theta}_j$ and then explore the profile on each side until the threshold ℓ_c is reached or an alternative stopping condition is met (e.g. for the profile of parameter j a bound $\in \{a_j, b_j\}$ is reached). Derivatives of the profile function may be used to inform the step size at each iteration [29, 66]. Software that implements this approach includes the *MATLAB Data2Dynamics* toolbox [29, 59] and the *Julia* package *ProfileLikelihood* [20].

Integration-based approaches formulate a Lagrangian set of equations from the log-likelihood objective and constraint in Equation 30 as a system of differential equations, which are then integrated [67–69]. This approach can be more efficient than step-wise approaches [68]. Software that implements this approach includes the *MATLAB Parameter ESTimation TToolbox (PESTO)* [70] and the *R dMod* package [71].

The constrained optimisation approach is based on a similar set of equations to the integration-based approach [22]. However, instead of formulating the set as a system of differential equations

it formulates them as an objective and set of constraints which are combined into a single function using the Augmented Lagrangian algorithm [72, 73] from the *C* package **NLopt** [34]. This approach is implemented in the *Julia* package **LikelihoodProfiler**. This methodology could likely be extended straightforwardly into two-dimensional interest parameter space when considering a 1-D line in that space.

Often, the profile likelihood's shape is also of interest, such as when considering the identifiability of parameters at many confidence levels. In this case, the step-wise approach, while potentially slower than the other approaches, has some merit.

2.3.4 Estimating Likelihood-Based Confidence Sets For Parameters

The PWA workflow introduces the idea of forming approximate likelihood-based confidence sets for the full parameter vector and an interest subset of the full parameter vector (an interest parameter vector). These parameter confidence sets are then propagated forward into confidence sets and reference tolerance sets for predictive quantities. Ideally, we would compute these parameter sets with an infinite number of points, such that all the uncertainty at that confidence level is captured in the prediction space. However, this is unrealistic in practice due to finite computational resources. Additionally, it may be the case that only a specific portion of the confidence set is required to capture the uncertainty at that confidence level fully.

Instead, we form an approximation of the confidence set using a finite number of points. For example, in Simpson and Maclaren [1] a finite number of points on the boundary of bivariate profiles at 95% confidence level is used to estimate the extrema of the profile-wise trajectory confidence set. They state that it is not necessarily valid to assume that the boundary of the parameter confidence set determines the extrema of the profile-wise trajectory confidence set; additional samples within the boundary may be required. This will be tested in Sections 5 and 6.

These ideas are closely linked with level set and excursion set estimation. A level set is defined as the set of parameter values for which some function is equal to a fixed value $\{f = c\}$ [74]. Similarly, the excursion set is the set of parameter values for which $\{f \geq c\}$ [74]. Considering the previously defined parameter confidence sets in Equations (12) and (14) we can see that the parameter confidence set is an excursion set where $\{f \geq \ell_c\}$ and the boundary of this set is a level set $\{f = \ell_c\}$. Level sets are, therefore, the confidence contours of the log-likelihood and profile log-likelihood functions. Our previous bivariate example is a case where we form an estimation of the 95% confidence level bivariate parameter level set using a finite number of points. Similarly, the confidence intervals in Section 2.3.3 are the $1 - \alpha$ univariate parameter level set. We will refer to the $\{f = \ell_c\}$ level set of a

parameter confidence set as its boundary for simplicity.

We note that we focus on finding points that cover a parameter confidence set defined by ℓ_c , regardless of their log-likelihood value above the threshold. This is in contrast to methods like Markov Chain Monte Carlo (MCMC) in a Bayesian context [7], which are trying to sample points proportionally to the posterior probability distribution (the log-likelihood when using uniform parameter priors).

2.3.4.1 General approaches

There are several general approaches to forming an approximate estimate of a parameter confidence set. In the context of likelihood evaluation, by defining parameter bounds $\theta \in [a, b]$, we can sample the parameter space randomly or by gridding and estimate the parameter confidence set by retaining samples for which $\hat{\ell}(\theta; y_{1:I}^o) \geq \ell_c$ (rejection sampling). This can be done using regular grids in parameter space [1] or through generating uniform random samples, Poisson Disk or Latin Hypercube samples [75]. The two key issues with rejection sampling approaches are the curse of dimensionality for increasing numbers of parameters and the sampling density failing to be informed by the likelihood function's value. A potential remedy could be using an initial coarse density to identify regions to sample more densely, but alternative methods may be more appropriate for identifying these regions regardless [76].

Improved approaches might instead seek to find a good approximation of the parameter confidence sets boundary first, which can then be used to define a sub-space to sample within. The bivariate profile boundary finding method in the workflow paper [1] is a simple example of this (see Section 4.1.1). Another example is the integral manifold approach, which finds the boundary exactly utilising the differentiability of structurally identifiable models [76] and is implemented in the *Julia* package **InformationGeometry** [23]. This method is primarily concerned with finding the boundary of the full parameter confidence set, not the lower dimensional profile-based confidence sets. Additionally, the implementation is currently limited to a maximum of three dimensions, requires a user-provided estimation of the error associated with observations, and requires structural identifiability of the model [23]. Bayesian level set estimation (LSE) approaches could also be used. LSE approaches use Gaussian processes to model the function space and identify optimal places to sample points from to improve the boundary estimate [74, 75, 77–79]. Both approaches could be of particular interest for future work that more efficiently identifies boundaries of our parameter confidence sets.

Other methods which use similar approaches to propagate uncertainty into sets for predictive quantities include Bootstrap methods [63], Randomised Maximum Likelihood (RML) Bootstrap [80], Metropolised RML [65], Bayesian Markov Chain Monte Carlo (MCMC) [7], sequential Monte Carlo

(SMC) [81] and approximate Bayesian computation (ABC) variants of rejection sampling, MCMC and SMC [81–83].

Implementations of the estimation of these sets include the approaches used for parameter confidence intervals in Section 2.3.3.3. For sets of higher dimensions, a rejection sampling approach and a layered step-wise approach outward from $\hat{\psi}$ for bivariate profiles are implemented in the *Julia* package `ProfileLikelihood` [20]. Similarly, MCMC is implemented in the *MATLAB* toolbox `Data2Dynamics` [59].

2.3.4.2 Univariate Profiles

Understanding how to evaluate a univariate profile using sampling is conceptually straightforward. We can first find the boundaries of the univariate profile which define the confidence interval and then sample points within the boundary. Alternatively, we can sample points between the parameter bounds and keep those within the profile boundary. We can then use interpolation along the points found, given their function values, to find the approximate confidence interval locations [13].

2.3.4.3 Bivariate Profiles

Bivariate profiles can be evaluated using sampling with any of the methods discussed in Section 2.3.4.1. Rejection sampling methods are easy to implement but will suffer from high rejection rates (low sampling efficiency) if the interest parameter boundaries are not well informed, such that the area of the bivariate profile is small relative to the area being sampled. As in the PWA workflow paper [1], here we consider methods (heuristics) for finding an approximation of the boundary of the bivariate profile. Once this approximation is found as a n -sided polygon, it can be used to sample points within the boundary efficiently using a rejection approach. We first generate a set of samples in interest parameter space, ensuring they are inside the polygon, and then reject them if their profile log-likelihood function value is less than $\ell_c = -\Delta_{\nu, 1-\alpha}/2$.

2.3.5 Polygons and Point Clouds

To work with and generate bivariate profiles, we review literature on working with point clouds and representing these clouds as polygons.

2.3.5.1 Minimum Perimeter Polygon Problems

The minimum perimeter polygon (MPP) problem is an example of the travelling salesman problem (TSP) [84]. While, in general, the problem is NP-hard [84], our use case can be solved (approximately) optimally using a TSP heuristic. Our use case is that we have generated a point cloud that corresponds to the boundary of a bivariate profile but are unsure how to connect the points to form a polygon

representation of the boundary in the best way. Assuming that there is only one boundary (level set), we use a TSP heuristic implemented in the *Julia* package `TravelingSalesmanHeuristics` [85] to determine the connections between points that create the MPP. If there are multiple boundaries, and they are not contained within each other, this approach should still contain the corresponding bivariate profiles. However, it will also contain points not in the bivariate profile, decreasing sampling efficiency.

2.3.5.2 Convex Hulls

The convex hull of a point cloud is the smallest convex polygon that contains all the points in the set, where the line segment between any two points in the set is contained within the polygon [86]. We use the Graham scan method [86, 87] implemented in the *Julia* package `Meshes` [88] to create a polygon representation of the boundary of a bivariate profile.

2.3.5.3 Concave Hulls and Alpha Shapes

Bivariate profiles may be non-convex [1]. If this is the case, rejection sampling methods applied to a convex polygon hull will sample from areas that do not represent the location of the bivariate profile [89]. To better represent the area of a bivariate profile, given an initial point cloud, we can use a k -nearest neighbours approach called the concave hull algorithm [89], implemented in the *Julia* package `ConcaveHull` [90]. This algorithm is a heuristic that aims to produce a polygon that better describes the area enclosed by the point cloud. The parameter k , for the number of nearest neighbours, can be considered a ‘smoothing’ parameter, which increases the smoothness of the polygon boundary for higher values of k . Here, we will set the value of k using a heuristic. Solving this problem in computational geometry has also been discussed using alpha-shapes [91, 92], which requires a similar smoothing parameter.

2.3.5.4 Level Set Continuation, Star Domains and Polygon Pole of Inaccessibility

If we have an initial starting solution for the polygon representation of a bivariate profile’s boundary, we may wish to use that starting solution in conjunction with a method that describes the direction to search for the boundary. If the starting polygon is convex or a star domain (a weaker condition), there will be a point within the starting polygon that could be used to inform search directions that do not overlap with a different polygon boundary.

We can connect the idea of homotopy continuation [93, 94] with the idea of moving between two distinct level sets of the profile log-likelihood function. A homotopy exists if two continuous functions, $f(\theta)$ and $g(\theta)$ from the same topological space can be continuously deformed into the other [93], such

as:

$$h(\theta, t) = (1 - t)f(\theta) + tg(\theta), \quad (31)$$

where $t \in [0, 1]$ and $h(\theta, t)$ is a homotopy between the two functions.

Similarly, consider the function which defines the boundary of a bivariate parameter confidence set, $\hat{\ell}_p(\psi; y_{1:I}^o) = \ell_c = -\Delta_{\nu, 1-\alpha}/2$, as we vary the confidence level of the threshold by varying α . If $\hat{\ell}_p(\psi; y_{1:I}^o)$ is unimodal, then values of ψ on this boundary can be defined such that gradually moving from a lower threshold boundary, $\ell_{c,\text{lower}}$, to the ℓ_c boundary will represent a continuous deformation of one boundary into the other. One way of defining the relationship between ψ for varying ℓ_c would be the proportion of the total perimeter length anti-clockwise along a given boundary between any given ψ and the point in the positive horizontal direction from $\hat{\psi}$. This can be seen in Figure 4.

A starting solution polygon is a star domain if there is an internal point such that the line segments between the point and all polygon vertices are also within the polygon [95]. The directions of these line segments from this point may be a good candidate for searching for the desired level set. The polygon's centroid could be used as this point if the starting solution is convex. If the set is not convex but is still a star domain, the polygon pole of inaccessibility (the internal point furthest from the polygon boundary) [96, 97] may be a more appropriate point to use, as the centroid may not be contained within the polygon (e.g. if the polygon is L or doughnut shaped). However, it may not be a point that makes the polygon a star domain, as this is not the objective of the pole of inaccessibility.

2.3.6 Sets for Predictive Quantities

2.3.6.1 Trajectory Confidence Sets

There are at least four common approaches to creating confidence sets for model solutions [45]. These will also be referred to as trajectory confidence sets. These include propagating forward a parameter confidence set formed using the ellipse approximation (FIM), a Bayesian MCMC approach which samples from the posterior predictive distribution (giving credible intervals rather than confidence sets), a frequentist approach which directly targets predictions using prediction profile likelihood (PPL) [10] and an ensemble modelling approach [45, 98]. The PPL sets are pointwise and require each time point to be evaluated separately. However, an integration-based approach [11] can reduce the severity

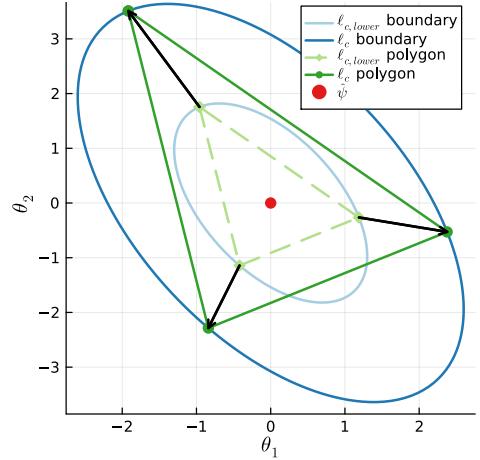


Figure 4: Continuation of a three point polygon on the $\ell_{c,\text{lower}}$ boundary, to the ℓ_c boundary.

of this requirement and improve computation time on ODE models, though this may be difficult to implement [45].

The approach discussed in this thesis (and Section 2.2.7) represents a fifth approach to creating trajectory confidence sets, $\mathcal{C}_{\phi,1-\alpha}$, which we take as defining a data distribution parameter, such as the mean. The value of our full trajectory confidence set is that it has curvewise (simultaneous) coverage guarantees. Additionally, once we have evaluated the full parameter confidence set, the full trajectory confidence set is straightforward to evaluate across many time points. Finally, we can also get curvewise full reference tolerance sets for observations, $\mathcal{C}_{y,(1-\delta,1-\alpha)}$, very cheaply after evaluating the full trajectory confidence set.

Evaluating full parameter confidence sets is often infeasible for higher dimensional models [1]. Resultantly, the PWA workflow instead presents the union of many univariate or bivariate profiles as a lower dimensional approximation to form approximate profile-wise trajectory confidence sets. This reduces the complexity of finding the parameter confidence sets from a problem in $|\theta|$ -dimensional space to up to $|\theta|$ one-dimensional problems or up to $\binom{|\theta|}{2}$ two-dimensional problems. These profile-wise trajectory confidence sets are significantly faster to compute and can reasonably approximate the full trajectory confidence set [1, 9]. Asymptotically, as the number of parameters in the interest parameter vector increases towards $|\theta|$, the profile-wise trajectory confidence set will converge towards the full trajectory confidence set. Providing a robust evaluation of the performance of these profile-wise trajectory confidence sets in terms of accuracy and computational time is a main outcome of this project. We will also consider ways that this performance could be improved.

Additional value can also be gained from the profile-wise approach; it allows visualisation of the impact of each parameter on different parts of the prediction space [1, 99, 100]. This can inform observation areas that could benefit from additional sampling to reduce uncertainty in these parameters and their propagated predictions. This is similar to using the profile likelihood of parameters to identify non-identifiable parameters and determine how to adjust the experimental design to resolve non-identifiabilities [7].

2.3.6.2 Observation Reference Tolerance Sets

As discussed in Section 2.2.8, we primarily focus on creating $(1 - \delta, 1 - \alpha)$ reference tolerance sets for the uncertainty quantification of observations. The practice of estimating reference intervals has roots in laboratory medicine and chemistry, where it is often of interest to quantify ‘normal’ ranges for values that impact health outcomes, such as serum cholesterol [18, 101, 102]. Here a ‘normal’ range refers to the values which would typically be expected in a population [18]. Tolerance intervals

provide a means of estimating a reference interval with a high degree of confidence [19, 101, 103] as their purpose is to “... locate the bulk of an underlying distribution of individual measurements” [25]. In other words, the purpose of a tolerance interval is to identify an interval that contains the reference interval with a high degree of confidence.

It is also asserted that tolerance intervals are more appropriate for the prediction of observations, at least in the context of mechanistic model and measurement error, because of the inclusion of a level of confidence [26, 104]. Tolerance intervals will have coverage of $1 - \delta$ of the population with confidence of at least $1 - \alpha$. This means that under repeated sampling at least $1 - \alpha$ of tolerance intervals formed are expected to contain the reference interval which contains $1 - \delta$ of the population. Additionally, because they are targeting the reference interval and not individual new observations, they can be re-used for multiple predictions. In contrast, prediction intervals will only have the correct coverage of $1 - \delta$ on average, with no confidence assurances on the probability of this occurring [26].

This focus contrasts the traditional process of creating prediction intervals for observations, motivating a discussion of the goal of forming prediction sets for observations. Some approaches, such as the validation profile likelihood (VPL) sets, which correspond to PPL sets [10], provide pointwise confidence sets for observations. However, pointwise statistics can be prone to issues of ‘bad averaging’ such as underestimating the extremes of epidemic curve ensembles [105]. Similarly, a Bonferroni correction, as used within the PWA workflow [1, 9], may produce pointwise sets that are overly conservative or result in significantly blowing up the size of prediction intervals if simultaneous guarantees across many time points are targeted. In the limit, as the number of time points approaches infinity, the simultaneous prediction set of observations would become a (100%, 95%) reference tolerance set, which is an effectively impossible goal. Additionally, these prediction sets can only be used for prediction once due to the nature of their coverage guarantees.

Therefore, a more reasonable goal for capturing observations, which stems naturally from our reference tolerance sets, could be capturing $1 - \delta$ of observations $1 - \alpha$ of the time rather than 100% of observations $1 - \alpha$ of the time. This presents a middle ground between pointwise statistics, which may underestimate extremes, and simultaneous statistics, which blow up in size. Additionally, because we are predicting the location of population reference sets, not individual observations, these sets can be reused for many predictions. The use of $(1 - \delta, 1 - \alpha)$ reference tolerance sets to meet this alternative prediction of observations goal will be evaluated in Sections 5 and 6.

2.3.7 Coverage Testing Confidence Intervals, Sets and Reference Tolerance Sets

We can perform repeated sampling experiments to test the frequentist property, coverage, of confidence intervals and sets, reference tolerance intervals and sets, and prediction sets [9]. For fixed time points, $t_{1:I, \text{train}}$, true parameters, θ , and the same mathematical model and error model, with a corresponding true model trajectory, $z(\theta^M)$, and $1 - \delta$ population reference set, we generate N data sets of training observations, $y_{1:I, \text{train}}^0$. Training observations are fixed across separate coverage simulations on the same model, allowing the comparison between simulations. We also generate N data sets of testing observations, $y_{1:I, \text{test}}^0$, at testing time points of interest, $t_{1:I, \text{test}}$. We then fit the interval/set of interest for each distinct data set and check whether the true value is contained within the interval/set (see the following sections). We then record the proportion of the N repeated experiments that contain the value of interest and compare this to the desired coverage level $1 - \alpha$.

If the methodology works well, we expect the proportion of successful experiments to be approximately $1 - \alpha$ or higher. Additionally, the uncertainty in coverage estimates will decrease as the number of simulations, N , increases. Confidence intervals for the coverage estimate are provided to quantify this uncertainty. The confidence interval for the estimated coverage is a Clopper-Pearson interval on a binomial test generated using the *Julia* package `HypothesisTests` [106].

2.3.7.1 Parameter Confidence Intervals

For parameter confidence intervals, we use the procedure in Section 2.3.3 to generate either Wald or likelihood-based confidence intervals for parameter j from each set of training observations. We then check if the true parameter value, θ_j , is within the confidence interval.

2.3.7.2 Bivariate Profiles

For profile likelihood-based confidence sets of bivariate interest parameters, we first create an approximate polygon representation of the profile boundary. We do this by combining the approaches discussed in Section 2.2.6 and 2.3.5 from each set of training observations. We then check if the true bivariate parameter value, $\theta_j \cup \theta_k = \psi$, has a profile log-likelihood function value greater than or equal to ℓ_c . If it does, we then check if ψ is contained with the polygon using a point in polygon algorithm, such as the algorithm implemented in the *Julia* package `PolygonInbounds` [107, 108]. This is equivalent to testing whether the true parameter value could be sampled from the polygon representation of the true boundary of the bivariate profile, subject to samples also being in the bivariate profile. As the polygon representation converges to the true level set boundary, the point in polygon algorithm tests if the samples are in the bivariate profile.

2.3.7.3 Full and Profile-Wise Trajectory Confidence Sets

For full/profile-wise trajectory confidence sets, we use the procedure in Section 2.3.6.1 to form confidence intervals for the model trajectory, $z(\theta^M)$, from each set of training observations. We will refer to this as either the coverage of full/profile-wise trajectory confidence sets or coverage of the model trajectory. We test the coverage of the model trajectory in a pointwise and simultaneous (or curvewise) manner. Both tests are useful, as the pointwise test can help with the diagnosis of reasons why the targeted simultaneous coverage was not reached [9]. Additionally, the pointwise test can demonstrate the dependency of different parts of the model trajectory on each parameter [1]. For pointwise tests, we form a confidence interval for the model solution at a fixed time point, $t_i \in t_{1:I,\text{test}}$ (note, not necessarily the time of an observation), and check if the true model trajectory $z_i(\theta^M)$ is inside this interval. For simultaneous tests, we test whether the true model trajectory is entirely contained within the full/profile-wise trajectory confidence set. In practice, this involves forming confidence intervals for the model trajectory for a fine grid of time points in a region of interest, $t_{1:I,\text{test}}$, and testing whether every interval contains the corresponding true model trajectory.

2.3.7.4 Full and Profile-Wise Reference Tolerance Sets and Observations

For full/profile-wise $(1 - \delta, 1 - \alpha)$ reference tolerance sets, we begin by forming the full/profile-wise $1 - \alpha$ confidence set for data distribution parameters from each set of training observations, as for trajectory confidence sets, using the profile-wise procedure in Section 2.3.6.1. We then propagate this confidence set into $1 - \delta$ reference set space using the procedure from Section 2.2.8. We will refer to this as the coverage of full/profile-wise reference tolerance sets or coverage of the $1 - \delta$ population reference set. We are testing that our full/profile-wise reference tolerance set contains the $1 - \delta$ population reference set with coverage of approximately $1 - \alpha$ or more. As for trajectory confidence sets (Section 2.3.7.3), we form these sets at a fine grid of time points in a region of interest $t_{1:I,\text{test}}$. For pointwise tests of full/profile-wise reference tolerance sets, we test whether the population reference interval at each $t_i \in t_{1:I,\text{test}}$ is contained within the corresponding reference tolerance interval. For simultaneous tests, we test whether the population reference set at every time point, $t_{1:I,\text{test}}$, is contained within the full/profile-wise reference tolerance set.

For coverage of observations from full/profile-wise reference tolerance sets, we use the N sets of testing observations, which correspond to the N sets of training observations; the n th testing observation set is used for the reference tolerance set formed using the n th training observation set. For pointwise coverage of observations, we test whether the observation $y_{i,\text{test}}^o$ at each $t_i \in t_{1:I,\text{test}}$ is contained within the corresponding reference tolerance interval. For simultaneous tests, we test whether every testing

observation, $y_{1:I,\text{test}}^0$, is contained within the reference tolerance set. For our alternative simultaneous goal, we test whether at least $1 - \delta$ of the testing observations are contained within the full/profile-wise reference tolerance set.

2.3.8 Area Coverage Testing of the Polygon Approximation of the True Boundary of a Bivariate Profile

To test the area coverage of a 2D polygon approximation of the boundary of a bivariate profile, we can follow the same type of procedure as in Sections 2.3.7 and 2.3.7.2 for testing frequentist coverage. This is distinct from the confidence set coverage otherwise discussed. For fixed time points, $t_{1:I}$, true parameters, θ , and the same mathematical model and error model, we generate N data sets of training observations, $y_{1:I,\text{train}}^0$. We then fit the approximate polygon representation for each distinct training data set. We sample a large number of points within the interest parameter bounds, accepting those within the bivariate profile using a rejection sampling approach. We then check the proportion of these points within the polygon, again using a point in polygon algorithm. This estimate will be reasonable for a sufficiently large number of sampled points. For a sufficiently good polygon approximation of a bivariate profile's boundary, we expect this coverage estimate to approach 1.0. Here, we will focus on the 0.05 quantile returned by the coverage simulation, meaning that 95% of the time, we expect coverage to be at least the reported statistic.

3 Computing Univariate Profiles

This section considers a straightforward approach for computing univariate confidence intervals using a step-wise approach, as initially discussed in Section 2.3.3.3. The intention is to produce an approach that is efficient enough to demonstrate the value of the PWA workflow. To compute the rest of the univariate profile, we will evaluate linearly spaced points inside this interval, as initially discussed in Section 2.3.4.2. The computation of these confidence intervals will also enable the assessment of parameter identifiability within the interest parameter’s bounds. We will also consider Wald confidence intervals and methods for improving the efficiency of our confidence interval approach. We explore the coverage properties of these univariate confidence intervals on three models that have been previously used to demonstrate the PWA workflow [1, 2], using the procedure discussed in Sections 2.3.7 and 2.3.7.1. Additionally, if the desired coverage is not observed, we investigate why this has occurred. Finally, we will compare the performance of our step-wise approach for finding parameter confidence intervals to the constrained optimisation approach of `LikelihoodProfiler`’s CICO method.

3.1 Parameter Identifiability and Finding Univariate Confidence Set Boundaries

We assess the identifiability of each parameter by evaluating its confidence set boundary (the confidence interval). This follows from the definition of identifiability in Section 2.3.2, where a parameter is identifiable if it has a finite likelihood-based confidence interval inside that parameter’s bound. We use a straightforward step-wise approach as seen in Algorithm 1.

This algorithm assumes the function is unimodal and continuous. If the function is multimodal, other approaches, such as evaluating interest parameter values spaced between the parameter bounds, may be more appropriate. Additionally, we use the `find_zero` function from the *Julia* package `Roots` [109] to find the location of the confidence interval. `find_zero` is used by supplying a bracketing interval and a bracketing algorithm. The bracketing algorithm’s absolute tolerance, `atol`, is used to check convergence.

A valid bracketing interval is one where the sign of the function on each side of the interval is different (i.e. $f(c) \times f(d) < 0$). Given that the function is continuous and can be evaluated within the bracket, a bracketing algorithm guarantees that we will converge to the confidence interval location. This makes it ideal for within `LikelihoodBasedProfileWiseAnalysis`, which is designed for use on many different models. Assessment of an appropriate bracketing method to use will be explored in Section 3.1.3.1. However, the convergence of the bracketing method is only guaranteed if the numerical optimisation scheme can correctly evaluate the sign of $f(\psi)$ for every value of ψ within the bracket. Ideally, the optimisation scheme would be able to evaluate the global maximum of $f(\psi)$ for every

Algorithm 1 Bracketing method for the boundaries of the univariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$, of parameter j , θ_j , which has a continuous unimodal profile log-likelihood function. Boundaries correspond to the likelihood-based confidence interval for θ_j . This example sets the confidence level of interest to 95%. Values of ω are the optimal values of the nuisance parameters, $\omega^*(\psi)$, given the most recent evaluation of the profile log-likelihood function.

Define the partitioning of parameters into interest parameter, $\psi = \theta_j$, and nuisance parameters, ω , with corresponding normalised profile log-likelihood function $\hat{\ell}_p(\psi; y_{1:I}^o)$

Define $\hat{\theta}$ as the maximum likelihood estimate of θ

Initialise $\text{atol} = 1 \times 10^{-3}$

Initialise $\psi_{\text{bounds}} = [a_j, b_j]$, $\hat{\psi} = \hat{\theta}_j$, $1 - \alpha = 0.95$, $\ell_c = -\frac{\Delta_{\nu,1-\alpha}}{2}$, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \emptyset$

Initialise confidence interval $\psi_{1-\alpha,[1,2]} = [\text{NaN}, \text{NaN}]$, identifiability value $\mathcal{I}_j = 0$

Define a new function which is zero at the threshold of interest $f(\psi) = \hat{\ell}_p(\psi; y_{1:I}^o) - \ell_c$

for $k \in \{1, 2\}$ **do**

- Initialise $\mathcal{F}_k = f(\psi_{\text{bounds},k})$, with optimised nuisance parameters $\omega^*(\psi_{\text{bounds},k})$
- if** $|\mathcal{F}_k| \leq \text{atol}$ **then**

 - $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_{\text{bounds},k}, \omega^*(\psi_{\text{bounds},k}))$

- else if** $\mathcal{F}_k < 0$ **then**

 - $\psi_{1-\alpha,k} = \text{find_zero}(f(\psi), (\psi_{\text{bounds},k}, \hat{\psi}), \text{Roots.Brent()}; \text{atol} = \text{atol})$
 - Evaluate $f(\psi_{1-\alpha,k})$ to obtain values of $\omega^*(\psi_{1-\alpha,k})$
 - $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_{1-\alpha,k}, \omega^*(\psi_{1-\alpha,k}))$

- else**

 - $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_{\text{bounds},k}, \omega^*(\psi_{\text{bounds},k}))$
 - $\mathcal{I}_j = \mathcal{I}_j + 1$

- end if**

end for

if $\mathcal{I}_j == 2$ **then**

- θ_j is structurally or practically non-identifiable at confidence level $1 - \alpha$ within ψ_{bounds} .

else if $\mathcal{I}_j > 0$ **then**

- θ_j is practically non-identifiable at confidence level $1 - \alpha$ within ψ_{bounds} .

else

- θ_j is practically identifiable at confidence level $1 - \alpha$.

end if

value of ψ , but only the weaker condition is required. If this does not occur, a `ConvergenceFailed` error will be thrown by `find_zero`.

3.1.1 Parameter Identifiability

We can assess the identifiability of each parameter at confidence level $1 - \alpha$ within the specified bounds by considering whether the bracket $(\psi_{\text{bounds},k}, \hat{\psi})$ is a valid bracket. By construction, $f(\hat{\psi}) = \ell_c > 0$. Resultantly, we only need to test the other side of the bracket. If either side of the interval is outside the specified bounds, we set it to `NaN` (non-identifiable). In this case, we use the corresponding value of ψ_{bounds} as that side of the interval for the univariate parameter confidence set.

If $f(\psi)$ evaluated at both sides of the interest parameter bounds $f(\psi_{\text{bounds},k}); k \in \{1, 2\}$, is less than zero (or less than `atol`) then that parameter is identifiable. If it is less than zero on only one side, then

that parameter is practically non-identifiable within the interest parameter bounds. Otherwise, it is either structurally or practically non-identifiable at this confidence level within the interest parameter bounds [22]. We cannot make any further identifiability statement without additional information. This is because we cannot separate the following impacts: the specified bounds preventing exploration of the correct region and that region not existing because the parameter is structurally non-identifiable. In this above case we instead determine visually whether the parameter is structurally or practically non-identifiable by evaluating the profile over a uniform grid of values of ψ informed by the location of the parameter bounds.

During profile visualisation we can also consider points (an additional width) outside the found interval to better understand the profile log-likelihood function. Additionally, the grid of points inside the confidence interval may be valuable for propagating our univariate parameter confidence set forward into sets for predictive quantities.

3.1.2 Likelihood-based Confidence Intervals and Wald Confidence Intervals

Before assessing the performance of the described algorithm, we briefly consider the consistency of the ellipse approximation of the profile log-likelihood function (Equation 27) with the profile log-likelihood function. For sufficiently regular likelihoods, the elliptical approximation will be very similar to the profile log-likelihood function [24]. Additionally, we demonstrate the consistency between approximations formed analytically and using an optimisation-based approach. This will be shown using the logistic model with a Gaussian data distribution from [1] (Appendix A.1). The model uses synthetic data. The profile log-likelihood function (both the ellipse approximation and actual function) for each parameter in this model, under a particular data realisation, is shown in Figure 5.

For the first parameter, the likelihood-based profile is skewed to the right relative to the approximate profile when moving away from $\hat{\theta}$. Resultantly, the Wald confidence interval for λ is shifted to the left relative to the likelihood-based interval. In contrast, the second parameter shows a very high level of consistency, indicating that the profile log-likelihood function is very regular along this interest parameter. Resultantly, for the parameter K , the likelihood-based and Wald confidence intervals are very consistent. The third parameter's profiles are broadly consistent on their right-hand side, with the likelihood-based interval's endpoint slightly higher. However, on the left-hand side, the Wald interval's endpoint is non-identifiable within the parameter bounds ($a_3 = 0$), while the likelihood-based interval's endpoint is identifiable.

The reasonable consistency between the Wald confidence intervals and their likelihood-based counterparts means they could be used as an initial starting solution to decrease the size of the bracketing

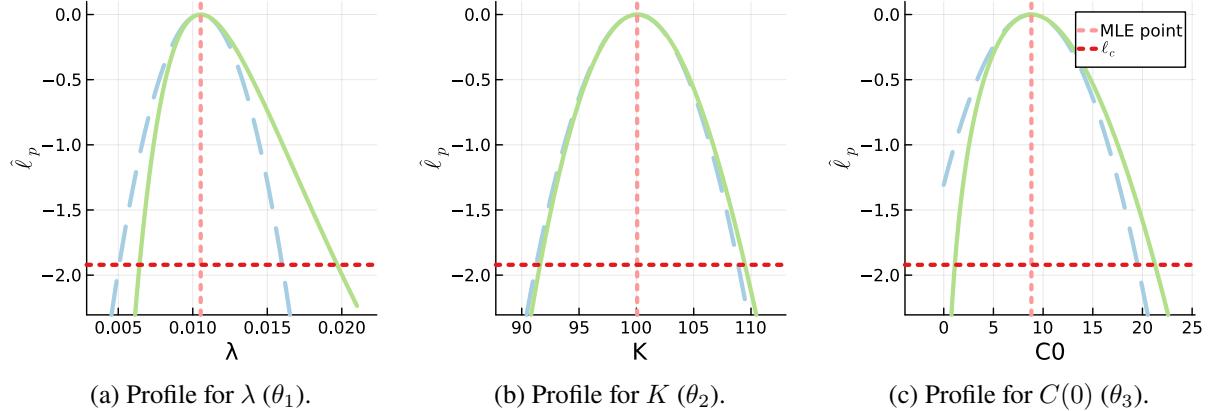


Figure 5: Normalised profile log-likelihood function (solid green) and ellipse approximation of that function (dashed light blue) for each parameter in the logistic model [1]. The vertical line (dotted light red) is the MLE value of each parameter. The horizontal line (dotted dark red) is the value of ℓ_c that corresponds to 95% likelihood-based confidence intervals for each parameter (intersection with the green line; intersection with the dashed blue line is the Wald confidence interval).

interval. The Wald confidence intervals can be found (effectively) for ‘free’ after a one-time evaluation of the hessian of the log-likelihood function at the MLE, $\mathcal{H}(\hat{\theta})$, using automatic differentiation (Section 2.3.1). Resultantly, using them to decrease the width of the bracketing interval may improve overall performance. This would be particularly useful for a high number of parameters; the one-time cost could be spread across the cost of evaluating many confidence intervals. Section 3.1.3.4 investigates this potential performance improvement.

Updating the interest parameter brackets is implemented by first testing if the each endpoint of the Wald interval is within ψ_{bounds} . For each endpoint that is within ψ_{bounds} we evaluate the function $f(\psi)$. If this function is less than or equal to atol , then we replace the left side of the bracket, $\psi_{\text{bounds},k}$, with that endpoint. Otherwise, we replace the right side of the bracket, $\hat{\psi}$, with that endpoint.

3.1.2.1 Ellipse Approximation Consistency: Analytical and Optimised

The analytical approach used to construct the approximate profiles (Equation 27) is directly connected to the nuisance parameter optimisation approach used for the profile log-likelihood function. Forming the approximate profile using the full Hessian matrix $\mathcal{H}(\hat{\theta})$ from Equation 25 (Section 2.3.1.1) is found to be equivalent to the analytical approach, as long as parameter bounds are not binding. This is seen for the approximated profiles in Figure 6; both approximations are identical except for the right endpoint of λ (Figure 6a). This lower parameter bound on the third parameter, $a_3 = 0$, prevents the optimised approximation from finding the ‘true’ optimal value on the approximated profile, which creates this inconsistency. If we increase this lower bound to $a_3 = 5$, we further constrain the optimised profile approximation, making the whole right side inconsistent with the analytical approximation, as

seen in Figure 7a. If we instead decrease this lower bound to $a_3 = -5$, we observe that the profile approximations from both approaches match, as seen in Figure 7b. These observations will also hold in higher dimensions.

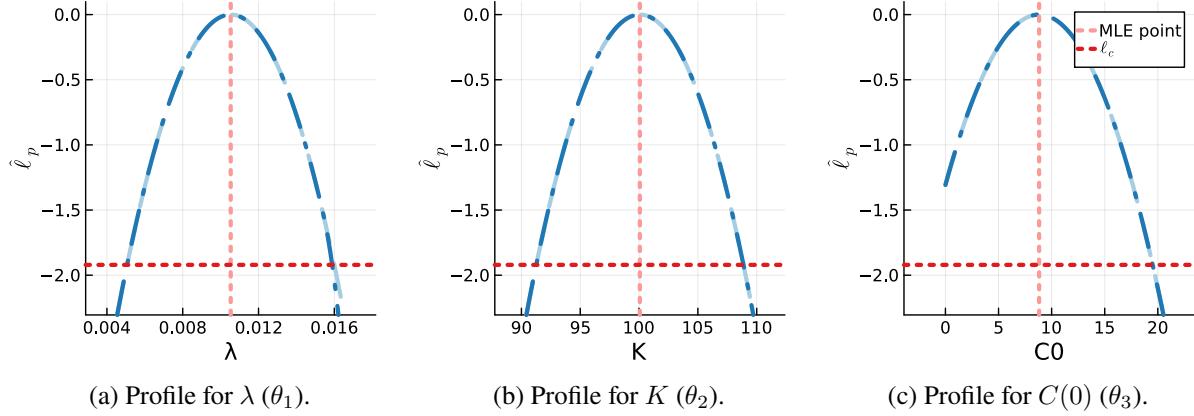


Figure 6: Analytical ellipse approximation of the profile log-likelihood function (dashed light blue) from Equation (27) versus the full ellipse approximation equation requiring optimisation of nuisance parameters (dash dotted dark blue) from Equation (25). For each parameter in the logistic model [1].

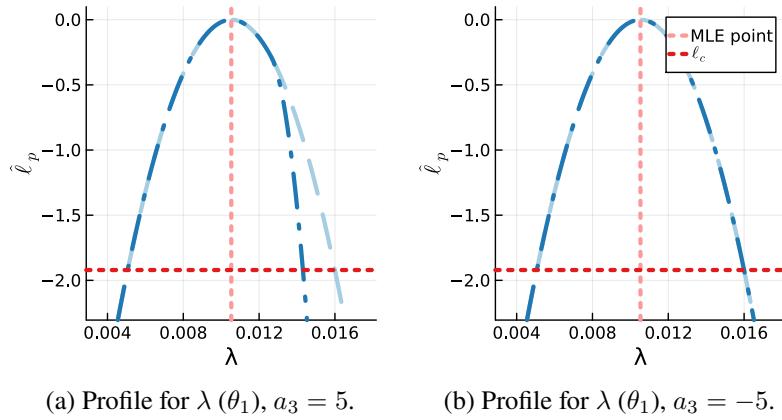


Figure 7: Analytical ellipse approximation of the profile log-likelihood function (dashed light blue) from Equation (27) versus the full ellipse approximation equation requiring optimisation of nuisance parameters (dash dotted dark blue) from Equation (25). For the first parameter in the logistic model [1], with the lower bound for the third parameter, a_3 , changed from zero to five, and zero to negative five.

This is a valuable demonstration to connect the analytical ellipse approximation with the profile log-likelihood function it is approximating. Additionally, it also provides a means of using Wald confidence intervals while enforcing feasibility constraints on parameters. Namely, the analytical Wald interval for λ assumes that the third parameter can be negative, whereas the optimisation approximation prevents this. Hence, approximate interest parameter confidence sets found using the optimisation approach may be more appropriate for propagation forward because of these feasibility constraints.

3.1.3 Algorithm Performance

In order to consider algorithm performance within our context, we explore the impact of various options on the number of likelihood function calls and optimisation function calls. We could also explore their impact on the algorithm’s time to complete. However, the models we consider are relatively simple, so implementation details may dominate raw timings instead of likelihood evaluations. Therefore, we will focus on function calls rather than computation time. This will allow our results to better generalise to more complex models where the cost of the likelihood evaluation will dominate the computation time over the cost of any algorithm implementation details (including programming language). This section will consider performance by applying the algorithm to the logistic model with a Gaussian data distribution from [1], as seen in Section 3.1.2.

A likelihood function call occurs when we evaluate the log-likelihood function, $\hat{\mathcal{L}}(\theta; y_{1:I}^o)$, for a specified value of θ . We are evaluating the log-likelihood function here, but we will regard the terms as equivalent for brevity. Similarly, an optimisation function call occurs when we evaluate the profile log-likelihood function, $\hat{\ell}_p(\psi; y_{1:I}^o)$, which requires many calls of the likelihood function to optimise out nuisance parameters at a specified value of ψ . Therefore, it is desirable to minimise optimisation function calls for best performance.

3.1.3.1 Bracketing Method Choice

In order to select an appropriate bracketing method, we compared the performance of methods implemented in Roots. We use the tolerance settings chosen in the subsequent section (Section 3.1.3.2). The performance of these bracketing methods on the logistic model example can be seen in Table 1, which records the mean number of function calls over the three parameters for a single data set. Tables featuring the information underlying Table 1 can be found in Appendix C.

We see the `FalsePosition`, `Brent` and `Chandrapatla` methods all perform very well in terms of their mean likelihood rank and mean optimisation rank. Therefore, any of these methods will be appropriate for our use. We choose `Brent`’s method [110] because it has the highest possible asymptotic efficiency (best-case convergence speed). `Brent`’s method is a hybrid method which combines the stability of a bisection method with the potential for faster convergence from a higher-order approach when appropriate. We are more interested in selecting a good method than the ‘best possible’ method, so we will not investigate this more robustly.

3.1.3.2 Impact Of Tolerance Settings

Algorithm performance is also impacted by the specification of the absolute tolerance in `find_-`

Method	Likelihood		Optimisation		Asymptotic Efficiency [109]
	Mean Rank	Mean Calls	Mean Rank	Mean Calls	
FalsePosition	1.67	1550	1.67	21.3	1.442
Brent	2	1640	3.33	24.3	≤ 1.89
Chandrapatla	2.67	1670	3.33	24.3	Not specified
ITP	5	1890	3.33	25.3	1.669
A42	5	2030	4	28.7	1.618
AlefeldPotraShi	6	2070	4.67	29	1.618
Bisection	6.67	2580	7	37	1.225
Ridders	7	2820	6.67	37.3	1

Table 1: The mean likelihood and optimisation rank, likelihood calls, optimisation calls and asymptotic efficiency of bracketing methods in `Roots` on the three-parameter logistic model [1], when finding the 95% confidence interval for each parameter. A method’s rank is calculated per parameter by considering the number of likelihood calls, with ties allowed. The optimisation rank is calculated in the same fashion. See Appendix C for more detail.

zero and the tolerance settings in the numerical optimisation algorithm. We will consider absolute tolerance and parameter relative tolerance within the optimisation algorithm. We will refer to these as `atol`, `abstol` and `xtol_rel`, respectively. The impact of separately increasing each of these from zero is considered in Figure 8 on the three-parameter logistic model [1]. We choose to use the absolute tolerance in `find_zero` rather than other settings so that convergence is not dependent on the magnitude of a given interest parameter.

Increasing `atol` from zero decreases the number of optimisation calls required for all parameter confidence intervals (Figure 8a). This is because the convergence criteria for `find_zero` becomes less strict, requiring fewer iterations to reach the point where $f(\psi) \approx 0$. This occurs after around $\log_{10}(\text{atol}) = -14$. Increasing `atol` also has the effect of decreasing the number of likelihood evaluations required, as fewer optimisation calls are needed. Therefore, we use $\log_{10}(\text{atol}) = -3$. The impact of this tolerance on the validity of the confidence threshold for one parameter is negligible: $\ell_c = -\Delta_{1,0.95}/2 = -1.9207$ at a 95% confidence level, while $\ell_c = -1.9207 - 0.001$ and $\ell_c = -1.9207 + 0.001$ represent a 95.006% and 94.994% confidence level, respectively. This will be even more negligible for thresholds from a higher parameter dimensionality, as ℓ_c increases for higher dimensions at the same confidence level.

Increasing `abstol` from zero significantly decreases the number of likelihood calls required for all parameter confidence intervals when $\log_{10}(\text{abstol}) \geq -14$ (Figure 8a). However, there are cases after this point where continuing to increase `abstol` increases the number of likelihood calls required. One of these cases is the jump from 1000 to 2500 likelihood calls for parameter two as $\log_{10}(\text{abstol})$

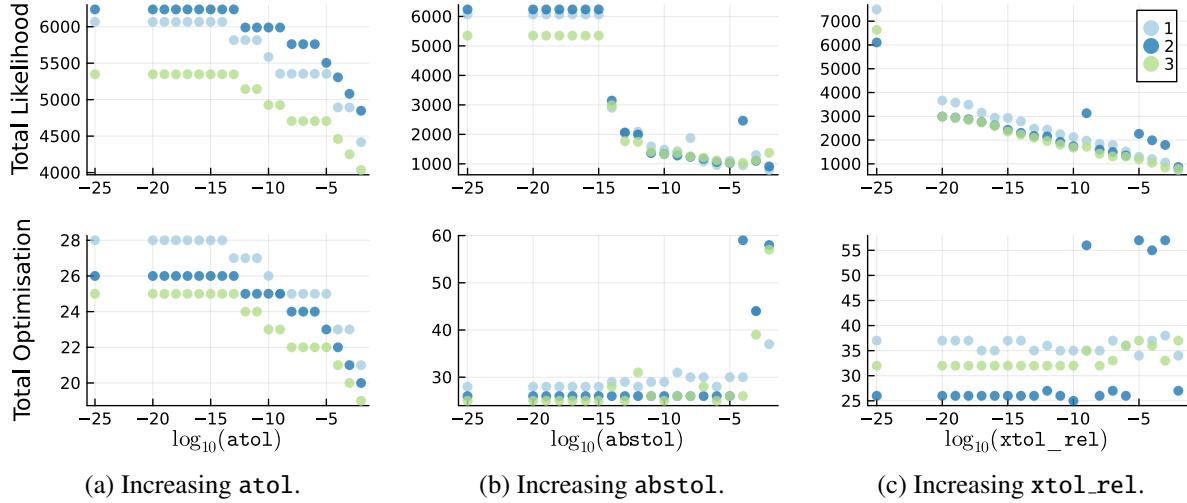


Figure 8: Impact of increasing `atol`, `abstol` and `xtol_rel`, holding the others constant at 0.0, on the total number of likelihood and optimisation calls required to find the 95% confidence intervals of the three parameters in the logistic model [1]. To allow visualisation of the number zero on these logged plots, we set $\log_{10}(0.0) \equiv -25$ (is $-\infty$ otherwise). Note: Figure 8c for `xtol_rel` was created using newer versions of some packages that have negatively impacted algorithm performance on parameters one and three (e.g. when all tolerances are 0.0); no other changes were made to the model. Nonetheless, all results can still be considered representative of the impact of using these settings.

increases from -5 to -4 .

The jump in likelihood calls occurs because increasing `abstol` increases the chances of getting stuck in a local minima or finishing prior to finding the global minima; `abstol` is the minimum required change in log-likelihood function value in each iteration of the numerical optimisation required for the scheme to continue searching. Resultantly, the correct value of $f(\psi)$ is more likely to be underestimated, which can drastically increase the number of optimisation calls required, as Brent's method relies on having (close to) the correct function value to choose its next point. That jump in likelihood calls for parameter two occurs because the number of optimisation calls doubles from 27 to 59. We will use `abstol` = 0.0 unless otherwise stated, noting that performance can be drastically improved from its use.

Finally, increasing `xtol_rel` from zero at least halves the number of likelihood evaluations required when $\log_{10}(\text{xtol_rel}) \geq -20$, while keeping the number of optimisation calls stable until $\log_{10}(\text{xtol_rel}) \geq -9$. Again, this is because the convergence criteria of the numerical optimiser becomes less strict, decreasing the number of likelihood evaluations. As for `abstol`, increasing `xtol_rel` increases the chances of getting stuck in local minima, and hence Brent's method may require a larger number of optimisation calls to converge. Unlike `abstol`, setting `xtol_rel` to very low values instantly improves performance without negative impacts, at least on this problem. Therefore,

we will use $\log_{10}(\text{xtol_rel}) = -12$ when evaluating the profile log-likelihood function in coverage simulations and for function performance comparisons. Unless otherwise stated, we will leave it at zero when finding the maximum likelihood estimate.

3.1.3.3 More Efficiently Finding Confidence Intervals at Multiple Levels

Suppose we wish to find Wald or likelihood-based confidence intervals for parameters at multiple confidence levels. In that case, we can use knowledge of the same type of interval (i.e. likelihood-based for likelihood-based) at a higher or lower confidence level to update the bounds used to bracket for each side of the interval of interest. If a higher confidence interval is known, it will enclose the interval of interest; these values can be used instead of the user-specified parameter boundaries, a_j and b_j . If a lower confidence interval is known, it will be contained in the interval of interest; these values can be used instead of $\hat{\psi}$. This can improve the performance of our bracketing method by reducing the search range. This will be particularly useful if the range of the initial parameter bound is large relative to the confidence interval range (i.e., it is not well-informed). The impact of knowing a lower or higher interval or both when evaluating likelihood-based confidence intervals, is considered in Figure 9.

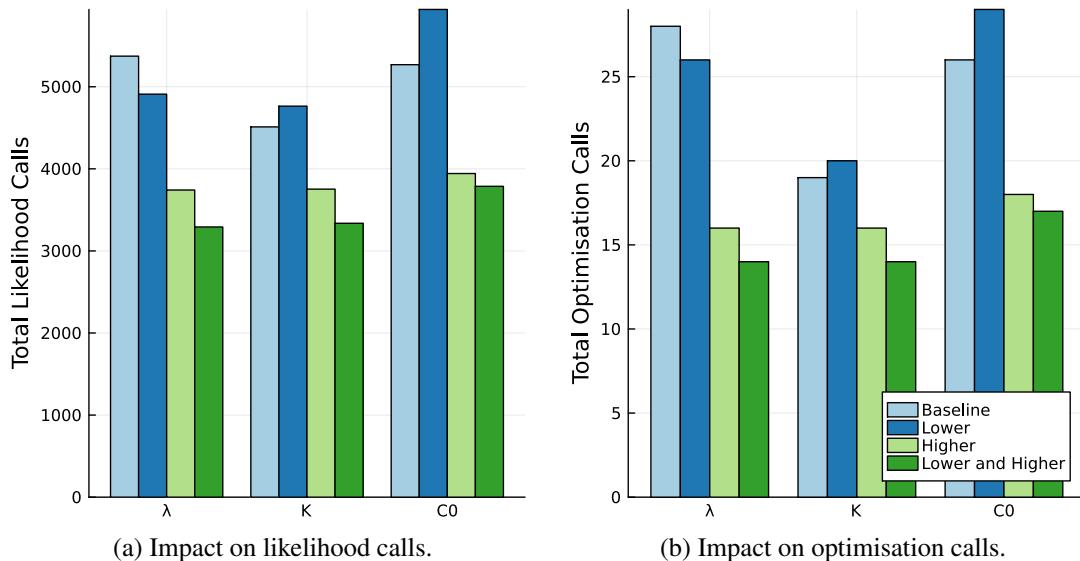


Figure 9: Impact of using a lower, higher or lower and higher confidence interval to reduce the bracket range versus not using them (baseline) on the total number of likelihood and optimisation calls required to find the 95% confidence intervals of the three parameters in the logistic model [1]. The lower and higher intervals used are the 90% and 99% confidence intervals, respectively. Initial parameter bounds are $a = [0.0, 50.0, 0.0]$, $b = [0.05, 150.0, 50.0]$, with 95% confidence intervals shown in Figure 5.

The story told by the likelihood call and optimisation call plots is very similar, so we will focus on the optimisation call plot (Figure 9b). Knowledge of only the higher confidence interval causes a significant

decrease in the total number of optimisation calls versus the baseline for all three parameters (28 to 16, 19 to 16 and 26 to 17, respectively). This is on average a decrease of 31% ($(43\% + 16\% + 35\%) / 3$). Knowledge of the lower and higher confidence interval causes these numbers to fall by several more calls. In contrast, knowledge of only the lower confidence interval either barely decreases the number of calls (for λ) or slightly increases the number of calls (for K and $C(0)$). These results indicate that if we wish to find confidence intervals for each parameter at multiple levels, they should be found from highest confidence to lowest. This ordering will generally allow the most drastic decreases in bracket range for subsequent confidence levels, particularly if a and b are not well informed and, therefore, the most drastic decreases in the number of likelihood calls. It is possible that there is a difference in ‘startup’ cost between finding the highest or lowest interval. However, we would expect the difference in results found here to dominate over any startup cost when finding several subsequent confidence intervals.

3.1.3.4 Using Wald Confidence Intervals To Reduce The Search Bracket

The impact of using the Wald confidence interval (found analytically) to update the bounds used to bracket for each side of the interval of interest (Section 3.1.2) can be seen in Figure 10. We noticed in Section 3.1.3.3 that updating the side of the bracket bounds that is typically given by ψ can result in worse performance. As a result, we consider the impact of using the Wald confidence interval to update the bracket bounds in the cases where it will only replace the outer side or either side indiscriminately.

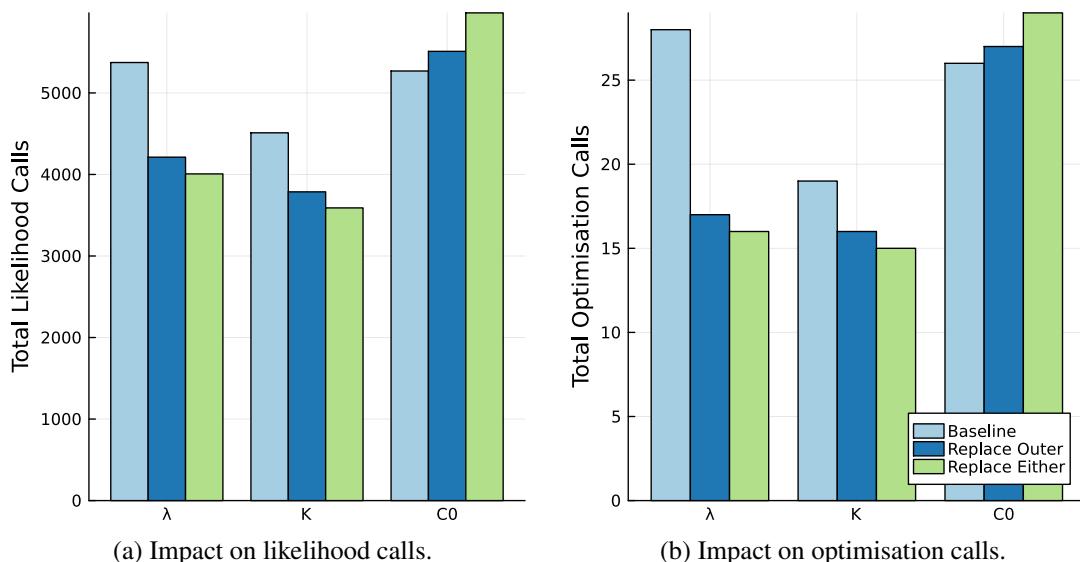


Figure 10: Impact of using Wald confidence intervals to reduce the bracket range (outer sides only or either side) versus not using them (baseline), on the total number of likelihood and optimisation calls required to find the 95% confidence intervals of the three parameters in the logistic model [1].

We find a similar result to Section 3.1.3.3 in Figure 10, where replacing the outer side of the interval

significantly reduces the number of likelihood and optimisation calls (for λ and K). Replacing either side instead of just the outer side improves performance a little more (for λ and K) or reduces it slightly (for $C(0)$). The result for $C(0)$ occurs because the left side of that Wald confidence interval is outside the lower parameter bound, and the right side of the interval is not on the outer side of the bracket (Figure 5c). Resultantly, only one additional optimisation call is incurred when checking whether the right side of the Wald interval is on the outer side of the bracket, explaining the increase from 26 to 27 in Figure 10b. The additional cost for using ‘replace either’ instead of ‘replace outer’ on $C(0)$ is offset by the performance gain for the other two parameters. The total number of likelihood calls across all parameters decreases from 15200 ($5373 + 4511 + 5269$) to 13600 ($4007 + 3590 + 5982$) and 13500 ($4214 + 3787 + 5510$), for the ‘replace either’ and ‘replace outer’, a decrease of around 11% ($100\% \times ((13500 - 15200) / 15200) = -11\%$). Therefore, using either approach would be appropriate. We will use the ‘replace either’ approach in all subsequent sections.

This approach may encounter issues or not improve performance at all for likelihood functions that are not sufficiently regular. In these cases, computation of the hessian, $H(\hat{\theta})$, may not be possible, preventing the calculation of Wald confidence intervals. Similarly, the hessian may be singular, such that the ellipse approximation of the profile log-likelihood function does not exist. Although we use a pseudo-inverse to prevent methods from breaking in this case, less is known about the validity of these Wald confidence intervals. Resultantly, using this approach may be a waste of computational effort in some cases.

3.1.3.5 Areas for Improvement

This algorithm could be improved in a few ways, notably by reducing the amount of duplicate optimisation function evaluations and using better initial guesses for parameters. The duplicate function evaluations arise due to using the `find_zero` implementation of bracketing instead of implementing a bespoke version. Additionally, within the bracketing method we do not save information on the profile likelihood function obtained from exploration within `find_zero`, unlike other step-wise approaches (Simpson and Maclaren [1] and Murphy, Maclaren, and Simpson [9] and `ProfileLikelihood` [20]). Better initial guesses could be used for nuisance parameters while bracketing to reduce the number of likelihood calls.

If we implemented our own bracketing method, we could avoid three extra optimisation function calls per identifiable side of the confidence interval (six total). This becomes a fourth extra call if we use the Wald interval to decrease the bracket size, and it replaces $\hat{\psi}$ in the bracket. This is reasonably significant; on the logistic model, Brent’s method required a mean of 24 optimisation function calls

per interval (Table 1) with a mean of 21 calls on another example in Appendix C. Resultantly, this implementation could reduce the required number of optimisation function calls by around 25% ($100\% \times ((18 - 24)/24) = -25\%$).

The duplicate calls occur for two reasons. Firstly, we evaluate \mathcal{F}_k to check that our bracketing interval is valid. However, `find_zero` also checks both sides of the bracket to prove the bracketing interval is valid. Both of these optimisation calls are not required as we know that $f(\hat{\psi}) = \ell_c > 0$ and we have already evaluated \mathcal{F}_k . The evaluation of $f(\hat{\psi})$ is not a huge deal, as the initial guess for nuisance parameters is their MLE values; convergence should be very quick. Secondly, we cannot guarantee that the final evaluation of $f(\psi)$ by `find_zero` is the ‘found’ location. Resultantly, we have to call the function again at the ‘found’ location to obtain values of ω . Therefore, implementing a bespoke Brent’s method is a valuable future contribution. Nonetheless, using `Roots` here was still helpful as it provided a straightforward way to check the performance of several bracketing algorithms on our problem (Section 3.1.3.1)

The other improvements (saving points explored while bracketing and nuisance parameter guesses) could also improve performance. However, we decided against these as, where possible, we have tried to keep the implementation simple and reusable across the package. Moreover, other methods such as `LikelihoodProfiler`’s CICO method [22] may be much more efficient (Section 3.3), so we did not try to over-optimize our heuristic. Initial testing was performed for the nuisance parameter guesses. However, we observed issues with convergence to the correct global maximum at each ψ . This potentially could have been resolved by using less naive guesses, e.g. the previous solution from the side of the bracket closest to the MLE is used as the guess.

3.2 Parameter Confidence Interval Coverage

This section will demonstrate the frequentist coverage properties of the likelihood-based confidence intervals on simple models explored by Simpson and Maclaren [1] and Simpson et al. [2]. We follow the procedure discussed in Sections 2.3.7 and 2.3.7.1 to test this coverage. The expected coverage properties are already shown in literature [24, 29]; we demonstrate it here to validate our implementation and the appropriateness of the PWA workflow on the models considered. We will focus on the coverage of 95% confidence intervals.

The models we will use are the logistic and Lotka-Volterra models with Gaussian data distributions from Simpson and Maclaren [1] and the two-species logistic model of coral reefs with real data (site 1 data is used) from Simpson et al. [2], which also uses a Gaussian data distribution. Model descriptions can be found in Appendix A, and visualisation of the univariate profiles from these models can be

found in Appendix D. Because the coral reef model uses real data rather than synthetic data, we will generate new synthetic data from the MLE model found when fitting to the real data to test coverage. The parameterisation, observation times and number of observations of these models are kept the same unless otherwise stated.

Considering the two-species logistic model, as in Simpson et al. [2], but in contrast to their statement in the text, the Nelder-Mead simplex algorithm [111, 112] will not be used as a local optimisation method. Based on the code available for the paper, the BOBYQA algorithm [35] is used as the main optimiser; it does not support a local optimiser, as it is itself a local optimisation method, so the specification of the Nelder-Mead algorithm does not work. The optimisation algorithm will also use smaller bounds for nuisance parameters; we use slightly wider versions of these here (Appendix A.3). We note that this impacts the interval estimated for two of the seven parameters to a large degree (parameters two and six, $\lambda_2, C_2(0)$) as seen in Appendix D.2. Four of the other five parameters remain unchanged, while the fifth changes a little (parameter three, δ).

3.2.1 Logistic and Lotka-Volterra Models

The parameter confidence intervals for both the logistic and Lotka-Volterra models show largely excellent coverage properties, with all parameters having simulated coverage within 0.02 of 0.95 in Figure 11. The 95% confidence interval for the coverage value given by all but one of the error bars contains 0.95. This confidence interval endpoint for $x(0)$ is only slightly lower at 0.946, which is not considered too low. Nonetheless, it may indicate that slightly more than 30 observations are required from the Lotka-Volterra model for the confidence intervals of simulated coverage to all contain 0.95.

3.2.2 Two-Species Logistic Model

In contrast, the coverage of parameter confidence intervals for the two-species logistic model is lower than expected in Figure 12a, with the coverage of only one parameter having a confidence interval enclosing 0.95 ($C_2(0)$ at 0.96 [0.95, 0.97]) However, $C_2(0)$ is likely to be non-identifiable on its left side within this simulation (see Appendix D.2), which results in the left side of the interval being considered the lower bound of 0.0. The ‘truth’ for $C_2(0)$ is 0.0293, so a large amount of the coverage on this parameter is likely because the left side of the interval was non-identifiable. Five of the seven parameters are around 0.91, and none have confidence intervals containing 0.95. Most significantly, the standard deviation parameter, σ , has a low coverage of 0.80 [0.77, 0.83]. This lack of coverage indicates an insufficient amount of or quality of experimental data, which prevents the likelihood function from being sufficiently regular. An insufficient amount of or quality of data was also indicated by the univariate profile plots obtained in Appendix D.2.

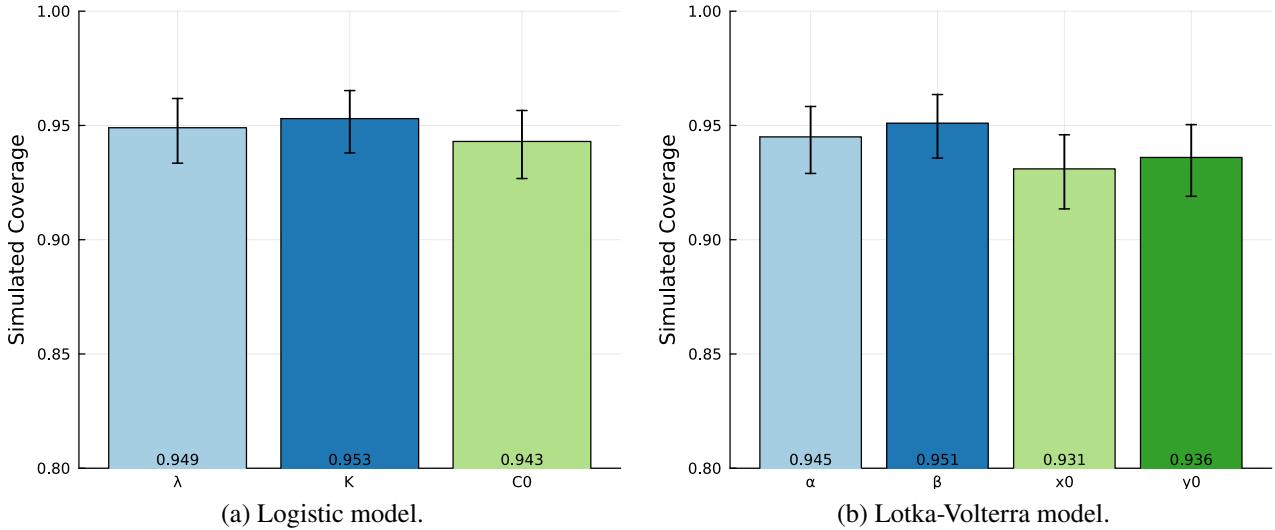


Figure 11: Coverage of 95% confidence intervals for each parameter in the logistic and Lotka-Volterra model [1] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

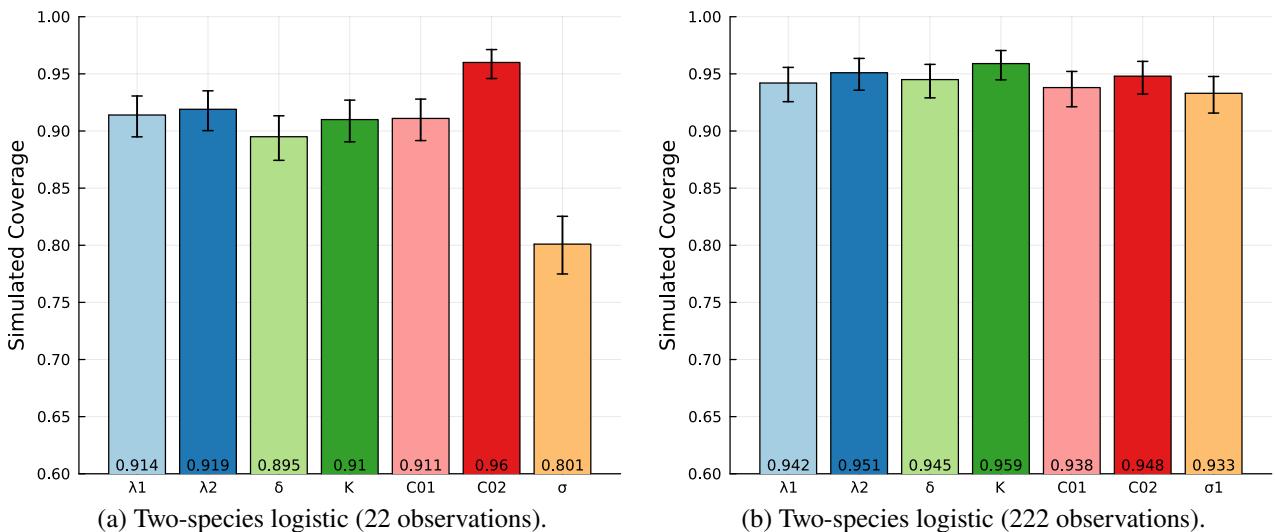


Figure 12: Coverage of 95% confidence intervals for each parameter in the two-species logistic model [2], with the same number of observations as in the data and additional observations from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars. Note: coverage ranges from 0.0 to 1.0; the increases are not as large as they appear between the two plots.

To test if more data would improve simulated coverage, we significantly increased the number of observation time points from 11 to 111 (total observations from 22 to 222). New time locations were chosen by placing ten linearly spaced points between each existing time point to keep results as consistent as possible (e.g. if original time points were [1, 11, 31], new points are [1, 2, 3, ..., 11, 13, 15, ..., 31]). As seen in Figure 12b, this fixes the coverage issues for all six problematic parameters. However, we note that the 95% confidence interval for the seventh parameter's coverage is only very close to

containing 0.95 ([0.916, 0.948]).

This is a valuable outcome of this coverage test. It can be used as a diagnostic tool to indicate whether a specified number of observations at a particular quality level is enough to obtain parameter confidence intervals with approximately correct coverage levels. It thus provides a means of quantifying the minimum amount of data to collect during an experiment if parameter confidence intervals are of interest. It could also be used after collection (e.g. using the $\hat{\theta}$ as the true parameterisation) as a sense check to indicate whether the coverage properties are expected to be approximately the correct level given the amount and quality of the collected data.

However, obtaining more data in this example may be unrealistic, given the time required to perform coral cover surveys. This could also motivate performing a recalibration of the confidence threshold, ℓ_c , so that our intervals produce the correct coverage. For example, we could use a bootstrap or Monte-Carlo procedure similar to Warne et al. [113] or Kreutz, Raue, and Timmer [10], respectively. This is left as future work. Instead, we emphasise that the asymptotic thresholds used for ℓ_c to estimate confidence intervals and sets are approximate for finite samples and are only guaranteed asymptotically [24]. It is reasonable to expect that we may have similar issues with the coverage of other sets on this example in later sections.

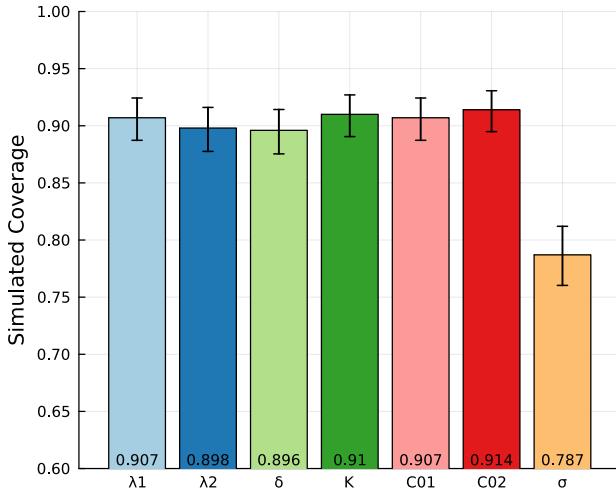
The convergence issues observed when creating the univariate profiles in Appendix D.2 indicate issues with the model used to represent the data. This is likely due to using a single additive Gaussian model as the data distribution, which is unrealistic for two reasons. Firstly, our data is a percentage defined on [0%, 100%], while the normal distribution is defined on $[-\infty, \infty]$. Secondly, the same standard deviation is used for the data from both coral types, while there is likely to be more ‘error’ in the larger observed percentages for $C_1(t)$ than the smaller observed percentages for $C_2(t)$. This motivates considering a more realistic data distribution. For example, the logit-normal distribution could be appropriate. It is used for proportions, defined on (0, 1) and expects the error to be larger around 0.5 and smaller close to 0 and 1. It cannot represent values of 0.0 and 1.0, but this is not an issue in this scenario. An observation of 0.0% would mean that a coral type has died out, which is inconsistent with the data. Similarly, an observation of 100% is exceptionally unlikely due to the long-time carrying capacity density, K , having an expected value of 81%.

3.2.2.1 Logit-Normal Data Distribution

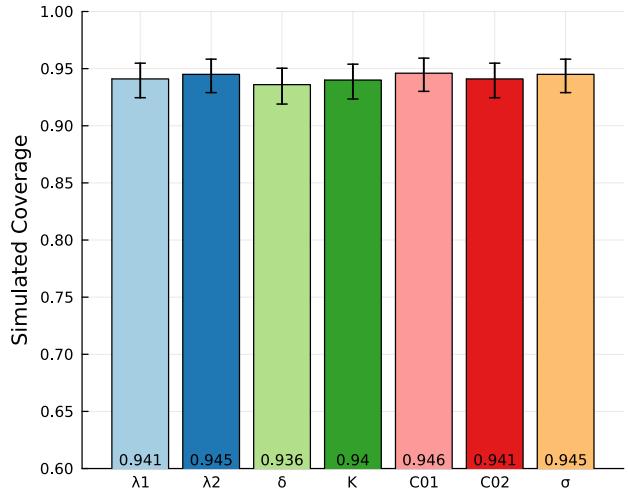
The improved univariate profiles for this example under the real data when using a logit-normal data distribution can be seen in Appendix D.2.1. Two signs of improvement from the data distribution change are the profiles’ smoothness and the more consistent match between the ellipse approximation

profiles and the profile log-likelihood function. Additionally, the new MLE estimate for the parameter $C_2(0)$, 1.19 instead of 0.0293, is also much more consistent with the observed data of 1.92. Computational performance was also significantly improved with the change; the total likelihood calls to find all confidence intervals improved from 160,000 to 70,000. Parameter bounds have been widened where relevant to allow each side of the confidence interval to be found (Appendix A.3.1).

For the coverage simulation, instead of using the MLE solution to simulate data, we choose a similar parameterisation, with $\theta = [0.003, 0.0004, 0.0004, 80.0, 0.4, 1.2, 0.1]$ (MLE values for θ given the real data were $\hat{\theta} = [0.00263, 0.00036, 0.000398, 81.3, 0.384, 1.19, 0.451]$). The results in Figure 13 are fairly consistent with those in Figure 12, including after significantly increasing the number of observations. This is not surprising, given that the simulated data is generated from whichever data distribution is being used; the low simulated coverage is a byproduct of too few observations for this model rather than model misspecification. Nonetheless, due to the other performance improvements of the model with the new data distribution, we will continue to use it with this model in later sections.



(a) Two-species logistic, logit-normal (22 observations).



(b) Two-species logistic, logit-normal (222 observations).

Figure 13: Coverage of 95% confidence intervals for each parameter in the two-species logistic model [2] using a logit-normal data distribution, with the same number of observations as in the data and additional observations from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

3.3 Performance Comparisons With LikelihoodProfiler

Here, we compare the performance in terms of the number of likelihood evaluations required to find both sides of the confidence interval of our methodology and the efficient CICO method in **LikelihoodProfiler** [22]. We use the defaults provided in **LikelihoodProfiler** with scan tolerances that allow finding the confidence interval for each parameter accurately to at least four significant figures. This parameter has a similar effect on the performance of **LikelihoodProfiler**

to our use of `atol` and `xtol_rel`. We note that we are using Wald confidence intervals to improve our method's performance, which is ignored in the performance cost calculation; the CICO method could also implement this to decrease its scan bounds and improve its performance. Additionally, we could improve the performance of our method by implementing a bespoke bracketing method, decreasing the number of likelihood calls by around 25%. If the percentage increase in calls between CICO and our method is significantly greater than 33%, then our method will be slower regardless of this improvement (i.e. decreasing the number of calls from 100 to 75 is a 25% decrease; similarly, 100 calls is a 33% increase from 75 calls).

`LikelihoodProfiler` also implements linear and quadratic extrapolation techniques similar to our method but with similar performance to CICO in the below examples. Their methodology is similar to ours, albeit using the optimal nuisance parameters from the current midpoint as the initial guess for the nuisance parameters of the next midpoint. On top of not making duplicate optimisation calls, this is likely to explain the performance gap between their extrapolation methods and our method. However, their linear extrapolation method saw convergence issues on the Lotka-Volterra model and failed to run, likely due to poor choices of nuisance parameter guesses. Our method tries to avoid this by only using the MLE as the initial guess.

3.3.1 Logistic

In the logistic example, we use a log transformation on the first parameter, λ , to enable the local Nelder-Mead optimiser used by CICO to perform well; it performs poorly for values around and smaller than 0.01. In order to compare it to our method correctly, we perform the same transformation. The lower bound on the transformed parameter is adjusted slightly away from zero prior to transformation (otherwise, the new lower bound would be $-\infty$).

The logistic model (Table 2) shows a very consistent estimation of the confidence intervals between the two methods. They all agree to at least three and usually four or more significant figures. CICO requires fewer mean likelihood calls for every parameter interval; our method, on average, requires 39.7% more likelihood calls $((57.4 + 1.2 + 60.5)/3)$. This is not a great result, but it is satisfactory, considering how much simpler our method is. It is also larger than 33%, indicating that implementing a bespoke bracketing method would still not result in our method having better performance than CICO on average.

	LikelihoodProfiler (CICO) [22]			LikelihoodBasedProfileWiseAnalysis			
Parameter	Lower	Upper	Mean Calls	Lower	Upper	Mean Calls	% Increase
λ	0.0064279	0.019686	895	0.00642679	0.019694	1410	57.4%
K	91.598	109.45	1110	91.597	109.47	1120	1.2%
$C(0)$	1.1172	21.261	967	1.1102	21.267	1550	60.8%

Table 2: Comparison of the performance of LikelihoodProfiler[22] and the method established in this section on the logistic model [1]. Compares the accuracy of finding parameter confidence intervals on a single data realisation and the mean likelihood function calls (3 s.f.) required across 100 data realisations.

3.3.2 Lotka-Volterra

The performance difference is stark on the Lotka-Volterra model (Table 3). Again, all parameters' confidence intervals are consistent to four or more significant figures. The mean number of likelihood calls required by CICO is lower for every parameter interval; our method, on average, requires 66.5% more likelihood calls ($(79 + 61 + 51 + 75)/4$). This is likely to be because the Augmented Lagrangian formulation at the heart of CICO [22] is more efficient at exploring through solution space, with better initial guesses for nuisance parameters, particularly with an extra parameter. It could also be partly due to the optimisation algorithms and settings chosen, although these are not expected to change the result significantly. This demonstrates that the CICO method is more appropriate than ours for finding parameter confidence intervals. It also indicates that this methodology could improve the performance of the bivariate boundary methods discussed in Section 4 if modified appropriately.

	LikelihoodProfiler (CICO) [22]			LikelihoodBasedProfileWiseAnalysis			
Parameter	Lower	Upper	Mean Calls	Lower	Upper	Mean Calls	% Increase
α	0.72118	0.96182	1540	0.72119	0.96182	2770	79%
β	0.99369	1.2342	1560	0.99368	1.2342	2510	61%
$x(0)$	0.69930	0.95973	1610	0.69930	0.95970	2430	51%
$y(0)$	0.17351	0.36847	1500	0.17350	0.36846	2610	75%

Table 3: Comparison of the performance of LikelihoodProfiler[22] and the method established in this section on the Lotka-Volterra model [1]. Compares the accuracy of finding parameter confidence intervals on a single data realisation and the mean likelihood function calls (3 s.f.) required across 100 data realisations.

3.3.3 Two-Species Logistic With Logit-Normal Data Distribution

The performance difference is quite stark on the two-species logistic model (Table 4). As in the logistic model, we use a log transformation on variables with values less than or around 0.01 for the CICO method (the first three parameters, $\lambda_1, \lambda_2, \delta$); we do not use this transformation for our method. Again, the confidence intervals for all parameters are consistent to three or more significant

figures, apart from the upper endpoint of $C_2(0)$. Both methods find that λ_2 and δ are non-identifiable at their lower endpoint. The mean number of likelihood calls required by CICO is considerably lower for every parameter interval; our method, on average, requires 168% more likelihood calls ($(177 + 112 + 129 + 146 + 308 + 172 + 129)/7$). This again demonstrates that the CICO method is more appropriate for use when finding parameter confidence intervals and could improve the performance of our bivariate boundary methods discussed in Section 4.

	LikelihoodProfiler (CICO) [22]			LikelihoodBasedProfileWiseAnalysis			
Parameter	Lower	Upper	Mean Calls	Lower	Upper	Mean Calls	% Increase
λ_1	0.002129	0.003222	2330	0.002122	0.003223	6460	177%
λ_2	NI	0.0009728	3980	NI	0.0009731	8430	112%
δ	NI	0.001020	3810	NI	0.001020	8710	129%
K	71.01	90.69	3020	71.01	90.69	7420	146%
$C_1(0)$	0.1801	0.8007	2060	0.1801	0.8007	8400	308%
$C_2(0)$	0.5511	2.374	2620	0.5511	2.390	7130	172%
σ	0.3448	0.6255	1140	0.3445	0.6256	2610	129%

Table 4: Comparison of the performance of LikelihoodProfiler[22] and the method established in this section on the two-species logistic model with a logit-normal data distribution [2]. Compares the accuracy of finding parameter confidence intervals on a single data realisation and the mean likelihood function calls (3 s.f.) required across 100 data realisations.

4 Efficiently Computing Bivariate Profiles

In this section, we discuss methods (math-heuristics) for finding fast and good approximations of the boundary of a bivariate profile, beginning with the method used in Simpson and Maclaren [1]. We identify a range of issues with this method; we seek to improve on these issues using an iterative process in which we propose several alternative methods. Instead of trying to find the most efficient method possible, our primary aim is to implement heuristics that are good enough to show the performance advantage of the PWA workflow conclusively. This will allow us to show the viability of using the approach and a framework with frequentist properties on the class of models discussed. Approaches for sampling points inside bivariate profiles' boundaries are further explored based on the initial discussion in Section 2.3.4.3.

The performance of the proposed boundary-finding methods will be evaluated in terms of their computational performance, coverage of the true boundary of bivariate profiles using the procedure from Section 2.3.8 and coverage of bivariate interest parameters using the procedure from Sections 2.3.7 and 2.3.7.2. Issues with performance in these areas will be explored and discussed. The optimisation function call performance will be evaluated using only the logistic model [1] used in Section 3. The other performance metrics will be evaluated using the three models used in Section 3. We will use the evaluation of performance in each of these areas to choose a method and settings for the three models in the following Sections 5 and 6, which explore the coverage of predictive quantities in the PWA workflow. We will also compare the computational performance of the chosen method against a rejection sampling approach.

4.1 Finding Profile Boundaries

4.1.1 Fix1AxisMethod

The simple method for finding bivariate boundaries from Simpson and Maclaren [1], as implemented in `LikelihoodBasedProfileWiseAnalysis`, is referred to as `Fix1AxisMethod`. Pseudocode for `Fix1AxisMethod` can be seen in Algorithm 2. While straightforward, it is not overly appropriate for use on more complex models due to several issues. We propose new heuristic methods by iteratively solving these issues with subsequent methods. In all methods, (unique) internal points found outside of `find_zero` function calls can also be saved within the bivariate confidence set $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$.

Algorithm 2 finds the desired number of boundary points by fixing the first and second interest parameters for half of these points each. It first draws a value for the fixed parameter, ψ_{x1} , using a uniform random distribution on provided bounds. Then, it draws two values for the unfixed parameter

Algorithm 2 Fix1AxisMethod for the boundaries of the bivariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$, of parameters j and k , $\theta_j \cup \theta_k$, which has a continuous unimodal profile log-likelihood function. Algorithm from Simpson and Maclaren [1]. This example sets the confidence level of interest to 95%. Values of ω are the optimal values of the nuisance parameters, $\omega^*(\psi)$, given the most recent evaluation of the profile log-likelihood function. `find_zero` searches for a zero along the 1D line between the two 2D points ψ_1 and ψ_2 (the method to do this is an implementation detail).

Define the partitioning of parameters into a bivariate interest parameter $\psi = \theta_j \cup \theta_k, j < k$ and remaining nuisance parameters ω , with corresponding normalised profile log-likelihood function $\hat{\ell}_p(\psi; y_{1:I}^o)$

Define $\hat{\theta}$ as the maximum likelihood estimate of θ

Initialise $\text{atol} = 1 \times 10^{-3}$

Initialise $\hat{\psi} = [\hat{\theta}_j, \hat{\theta}_k]$, $1 - \alpha = 0.95$, $\ell_c = -\frac{\Delta_{\nu,1-\alpha}}{2}$, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \emptyset$

Define a new function which is zero at the threshold of interest $f(\psi) = \hat{\ell}_p(\psi; y_{1:I}^o) + \ell_c$

for $z \in \{1, 2\}$ **do**

if $z == 1$ **then**

$\psi_x = \theta_j$, $\psi_{\text{bounds},x} = [a_j, b_j]$

$\psi_y = \theta_k$, $\psi_{\text{bounds},y} = [a_k, b_k]$

 totalPairs = totalPoints % 2

else

$\psi_x = \theta_k$, $\psi_{\text{bounds},x} = [a_k, b_k]$

$\psi_y = \theta_j$, $\psi_{\text{bounds},y} = [a_j, b_j]$

 totalPairs = totalPoints % 2 + totalPoints % 2

end if

for $p \leftarrow 1$ to totalPairs **do**

while true **do**

 Sample $\psi_{x1} \leftarrow \text{rand}(\psi_{\text{bounds},x})$, $\psi_{y1} \leftarrow \text{rand}(\psi_{\text{bounds},y})$, $\psi_{y2} \leftarrow \text{rand}(\psi_{\text{bounds},y})$

$\psi_1 = \psi_{x1} \cup \psi_{y1}$

$\psi_2 = \psi_{x1} \cup \psi_{y2}$

if $f(\psi_1) \times f(\psi_2) < 0$ **then**

break

end if

end while

$\psi_{1-\alpha,p} = \text{find_zero}(f(\psi), (\psi_1, \psi_2), \text{Roots.Brent}(); \text{atol} = \text{atol})$

 Evaluate $f(\psi_{1-\alpha,p})$ to obtain values of $\omega^*(\psi_{1-\alpha,p})$

$\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_{1-\alpha,p}, \omega^*(\psi_{1-\alpha,p}))$

end for

end for

in the same fashion (ψ_{y1} and ψ_{y2}). If one of these points ($[\psi_{x1}, \psi_{y1}]$ and $[\psi_{x1}, \psi_{y2}]$) is an internal point and the other an external point, the point pair is accepted as they are a valid bracket. Internal and external points are points inside ($f(\psi) > 0$) and outside ($f(\psi) < 0$) the confidence boundary, respectively. This is checked by testing whether $f(\psi)$ has opposite signs for each point. Brent's method within `find_zero` is then used to find a boundary point between the point pair. The method searches for valid point pairs until the desired number of boundary points is found.

The main issues with Algorithm 2 are as follows. Firstly, it has unnecessarily constrained search

directions, only searching along the x and y axes. Secondly, it is inefficient at sampling: it throws away both parts of a point pair if the pair does not bracket a boundary, and it does not reuse points that were hard to find (internal points will generally be hard to find if the parameter bounds are not well-informed), and it does not reuse or use prior knowledge of the bivariate profile space. Additionally, the impact of parameter bounds can cause inefficient sampling if the area of the confidence boundary is small relative to the parameter bounds. Thirdly, if parameter bounds are inside the profile boundary, it will never sample external points, looping infinitely. Fourthly, it does not guarantee good boundary coverage as it may only sample points on a particular part of the boundary and does not give info on where bounds get in the way. Finally, it does not provide a method for finding new internal points within the boundary, which may be important for correctly propagating uncertainty into prediction space.

4.1.2 SimultaneousMethod

`SimultaneousMethod` is a natural progression of `Fix1AxisMethod`, generalising the search directions and more efficiently sampling point pairs. Pseudocode for the `SimultaneousMethod` can be found in Appendix E.1 and E.1.1. Unlike `Fix1AxisMethod`, the `SimultaneousMethod` does not require pairs of points to be along the x or y axis or require a pair to be found using subsequent random draws. Instead, we draw a new point using a uniform random distribution within the bounds, check whether it is an internal or external point, and then add it to the relevant internal or external point queue. These points are then paired in the order found. These changes resolve the constrained search direction issue and significantly improve sampling efficiency by throwing away fewer points. The method uses four random numbers instead of the three consumed by `Fix1AxisMethod` to find each accepted point pair, but this impact is greatly outweighed by throwing away fewer points.

If finding internal or external points is difficult, given the specified bounds on interest parameters, we can reduce the proportion of unique internal or external points to improve sampling efficiency further. We can also use the MLE point as the first internal point. For example, consider the case where we want 100 boundary points and finding internal points is ten times more difficult than finding external points. In this example, we only find ten internal points by the time we find 100 external points. In order to find another 90 internal points, we would, on average, have to sample another 900 external points. Instead, we can reuse these internal points, pairing these 10 points with each set of 10 external points in the order found. This would decrease the number of sampled points (and thus optimisation function evaluations) from 1100 to 110, significantly improving sampling efficiency.

If a parameter bound is in the way of reaching the boundary, points will not be put on that bound.

Additionally, if the true boundary is very close to a parameter bound, the method will struggle to find this region of the boundary. This is because finding the boundary in this location requires generating a random point between the boundary and the parameter bound, which becomes more difficult the closer they are. The method, like `Fix1AxisMethod`, will infinite loop if the interest parameter bounds are inside the boundary.

4.1.3 RadialRandomMethod

`RadialRandomMethod`, in contrast to the previous two methods, only tries to find internal points via sampling, with their reuse of particular interest. Pseudocode for the `RadialRandomMethod` can be found in Appendix E.1 and E.1.2. Instead of finding external points via sampling, we define the number of radial directions (`num_radial_directions`) to search for the boundary from each point. The method then chooses `num_radial_directions`, spaced uniformly around a circle, and rotates these directions by a random phase shift $\in [0.0, 2\pi/\text{num_radial_directions}]$ radians. These directions are then distorted by the relative magnitude/scale of the two interest parameters (may be user-specified). Then, for each of these directions, until it runs out of directions or finds the desired number of points, it finds the closest point on the interest parameter bounds to the internal point in this direction. If the point on the bounds is inside the boundary, it will be used as the boundary point. If the point on the bounds is outside the boundary, it will be used as the external point in the point pair. These search directions have the positive side effect of giving better-expected coverage guarantees than the previous two methods, for $\text{num_radial_directions} > 1$, because we can guarantee that directions opposing each other have been explored. We can also use the MLE point as the first internal point to improve sampling efficiency and coverage guarantees further.

Given a fixed number of desired boundary points, we can decrease the cost of finding internal points by increasing the number of radial directions to explore, `num_radial_directions`, at each internal point. This improves sampling efficiency. However, it is important to balance `num_radial_directions` with the desired number of boundary points. If `num_radial_directions` is large relative to the number of boundary points, then the boundary the method finds may constitute a local search around the internal points. Resultantly, there may be regions where the boundary is not well explored. This effect can be seen in Figure 14, where many boundary points are found in the bottom right of the boundary, but very few elsewhere for `num_radial_directions = 8`. Decreasing this parameter to four or two dramatically improves the spread of these points, leading to an improved polygon approximation of the boundary. This will be even more of an issue for more concave boundary regions because local searches from a single internal point may not be able to ‘see’ all of the boundary.

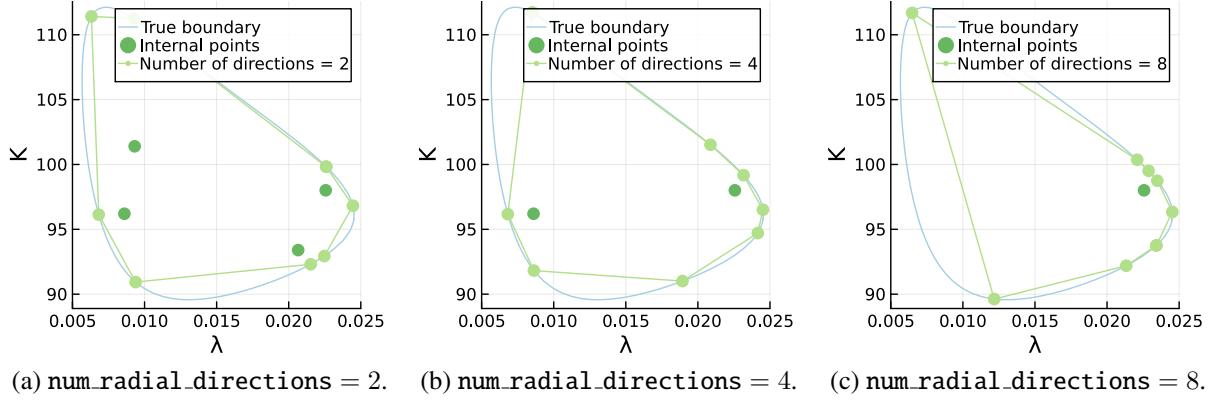


Figure 14: Eight points found using `RadialRandomMethod` on the 95% confidence set boundary for the first two parameters in the logistic model [1] with two, four and eight radial directions per internal point, respectively. The location of internal points is constant.

This method will not infinite loop if the interest parameter bounds are fully contained within the boundary, unlike the previous methods. This is because this method will use points on the bounds as boundary points if the bounds prevent reaching the boundary. Additionally, using these points as boundary points can be used to diagnose that the parameter bounds are overly restrictive or that the parameter is not identifiable in this direction at confidence level $1 - \alpha$.

4.1.3.1 On Logarithmic Reparameterisations

Search directions are distorted by parameter magnitude so that the relative scale of two parameters does not impact which points are found. This is part of the reason working in logarithmic parameter space can be advantageous, which is a common practice [22, 29, 52, 71, 114]). The impact of quite different parameter scales can be seen in Figure 15, where the undistorted search directions result in effectively only two unique points (of four) being found along the parameter with the larger scales axis. In contrast, the distorted search directions and logarithmic reparameterisation do not have the same issue.

Logarithmic reparameterisations are also straightforward to use within the framework and methods we provide here. However, we provide the relative magnitude distortion as a simple alternative to improve the performance of methods that use search directions. We provide this alternative because working in a logarithmic parameterisation can be non-ideal for sampling-based methods when parameters have a lower bound near zero, as log space will blow up near zero. Consequently, the relative area between the bivariate confidence set and the parameter bounds for the original and logarithmic parameterisation can be significantly different, making it much harder to find internal points in the logarithmic parameterisation. This can be seen in Figure 16 where the area of the 95% confidence boundary compared to parameter bounds is significantly larger in standard space than logarithmic

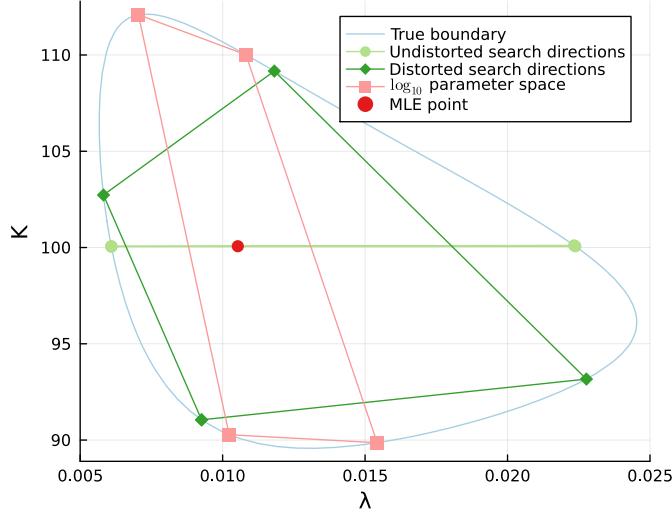


Figure 15: Impact of distorting search directions or transforming the parameter space when finding the bivariate boundary from parameters one and two in the logistic model [1]. Finds four boundary points from the MLE point using `RadialRandomMethod`. Parameter magnitude is 0.05 and 10 for parameters one and two, respectively, when using distorted search directions.

space because the lower bound on λ is effectively zero. Sampling-based methods are significantly less efficient when used with the logarithmic parameterisation for this profile.

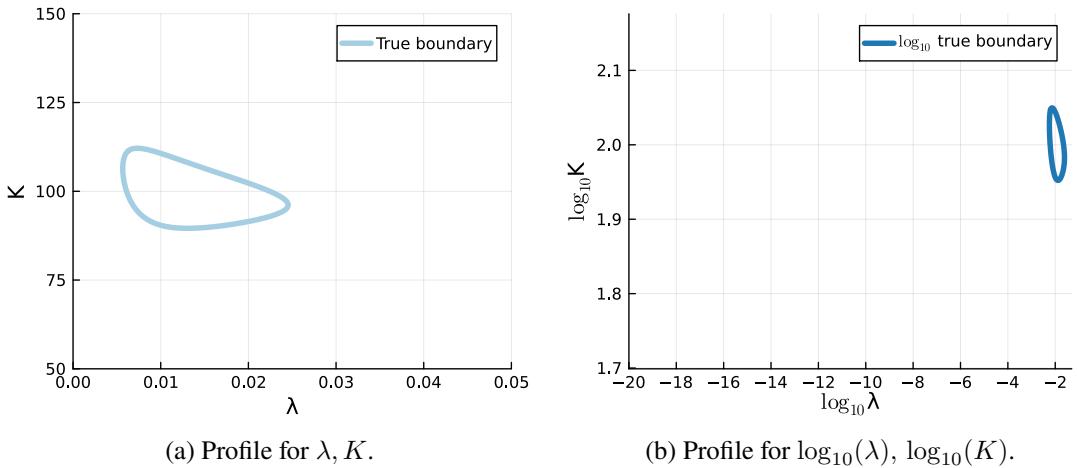


Figure 16: 95% confidence set boundary for the first two parameters in the logistic model [1] in comparison to the parameter bounds in standard and log-10 parameterisations.

4.1.4 AnalyticalEllipseMethod Using Julia Package EllipseSampling

For bivariate confidence profiles from the ellipse approximation of the profile log-likelihood function, $\hat{\ell}_p^E(\psi ; y_{1:I}^o)$, we can find the exact equation of the ellipse corresponding to the confidence threshold, ℓ_c , as mentioned in Section 2.3.1. We can then sample points on the ellipse's boundary with respect to arc length using equal or clustered spacing. The methodology required for this is implemented in the author's *Julia* package `EllipseSampling` [4] and is covered here. By normalising Equation (28)

by ℓ_c so that one side of the equation is equal to 1, we obtain the equation of an ellipse [42]:

$$\begin{aligned} 1 &= -\frac{1}{2\ell_c}(\psi - \hat{\psi})'([e_j, e_k]' \Gamma(\hat{\theta}) [e_j, e_k])^{-1}(\psi - \hat{\psi}), \quad \theta_j \cup \theta_k = \psi \\ &= (\psi - \hat{\psi})' \mathcal{C}(\psi - \hat{\psi}), \end{aligned} \tag{32}$$

where $\mathcal{C} = (-1/2\ell_c) \times ([e_j, e_k]' \Gamma(\hat{\theta}) [e_j, e_k])^{-1}$. The major and minor axis radii, a^E and b^E respectively, can then be evaluated by considering the inverse of the square roots of the eigenvalues of \mathcal{C} (ordered from largest to smallest). To determine the rotation, α^E , of the major axis of the ellipse from the positive x axis, we calculate the inverse tangent of the division of the y and x components of the eigenvector corresponding to the largest eigenvalue.

Now that we have the values of a^E , b^E and α^E , as well as the location of the centre of the ellipse, $\hat{\psi}$, we have everything needed to sample on the ellipse boundary. To enable easy sampling along the boundary of the ellipse, we disregard any rotation or translation; these will be applied after sampling points. We then sample along the ellipse's boundary with equal spacing or a clustered spacing with respect to arc length, where clustering occurs around the major axis vertices (i.e. the region of greatest curvature). Sampling occurs anti-clockwise.

The location of the first point on the ellipse is determined by a start point shift argument between zero and one. A value of zero (or one) means the first point is placed at the location of the most positive major axis vertex. Values greater than zero rotate this point anti-clockwise around the ellipse, travelling along the arc until that proportion of the total perimeter has been travelled.

4.1.4.1 Clustering Parameter Impact

Figure 17 shows the impact of the clustering parameter and the magnitude of the major and minor axis radii. The strength of the clustering is dependent on the relative magnitudes of the major and minor axis radii. If the major and minor axis have the same radii, then the ellipse is a circle and no clustering will be observed (Figure 17a). As the magnitude of the major axis radius increases relative to the minor axis radius, the strength of the clustering for the same clustering parameter value will increase (Figure 17b).

The clustering effect is beneficial when we use this initial boundary to guess a profile log-likelihood boundary. This guess will be reasonable when the profile log-likelihood function is sufficiently regular. Suppose we seek to find the new boundary by tangentially pushing out from the starting ellipse at each generated point. In that case, the points that diverge the fastest are located at the region of greatest curvature. Similarly, if we push out from the starting ellipse radially from its centre, we might expect the log-likelihood function to have a similar region of greatest curvature. Resultantly,

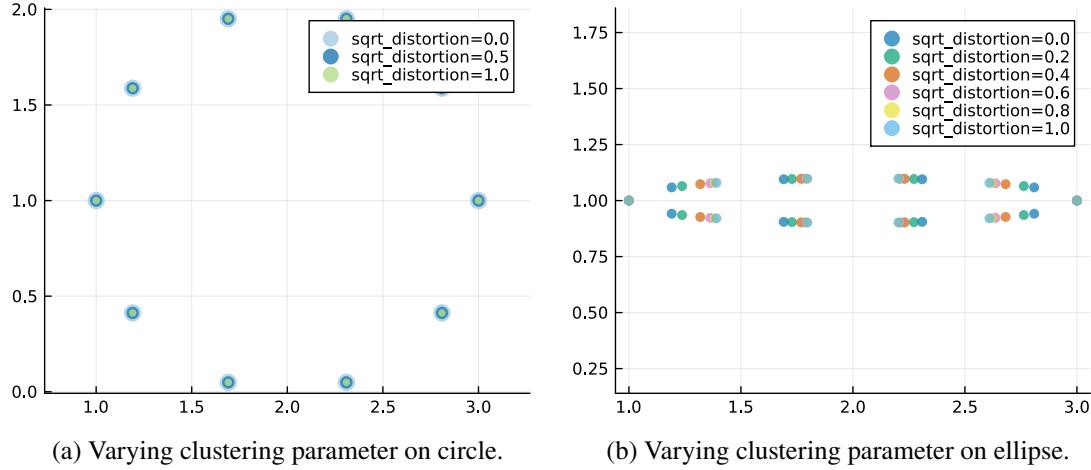


Figure 17: Varying the clustering parameter, ‘sqrt_distortion’, for 10 points placed on an ellipse with $(a^E, b^E) = (1.0, 0.1)$ and circle with $(a^E, b^E) = (1.0, 1.0)$. Both are centred at $[2, 1]$ and are not rotated. Plots from `EllipseSampling` [4].

when generated points are equally spaced, the new boundary is likely to be well-defined in regions that are approximately parallel to the major axis of the starting ellipse but poorly defined in all other regions. Thus, having the clustering effect allows the method to quantify the uncertainty in the polygon approximation of that curvature better.

4.1.5 ContinuationMethod

The `ContinuationMethod` was the first attempt at using the information of points already found and local approximations from the `AnalyticalEllipseMethod` to inform locations to explore for the bivariate confidence boundary. It performed poorly relative to the other methods from a computational and boundary exploration standpoint, so no pseudocode is provided. However, we discuss it here because the lessons learnt through its development and its core idea were central to the creation of the `RadialMLEMethod` and `IterativeBoundaryMethod`, which are covered in Section 4.1.6 and 4.1.7, respectively.

The core idea of the method is to form an equation which is similar to a homotopy map between the bivariate boundary at the desired confidence level and some lower confidence level, where $\hat{\ell}_p(\psi; y_{1:I}^0) = (1 - t)\ell_{c,\text{lower}} + t\ell_c$. Here, we assume that the profile log-likelihood function is continuous and unimodal. The solution (polygon) for the lower confidence level boundary (for $t = 0$) is then ‘continued’ to the higher confidence level boundary (for $t = 1$), hence the name. Given a sufficiently regular log-likelihood function, at a low enough confidence level, the points found using `AnalyticalEllipseMethod` will be a good local approximation of points on the $\ell_{c,\text{lower}}$ bivariate boundary. Therefore, we use these points to find the $\ell_{c,\text{lower}}$ bivariate boundary, searching from $\hat{\psi}$ towards each point until a point on the boundary is found. These points on the true $\ell_{c,\text{lower}}$ boundary

are then continued.

Continuation here means hopping from the polygon approximation of a lower confidence boundary to a higher confidence boundary in a discrete set of steps (e.g. $t = [0, 0.1, 0.2, \dots, 1.0]$). The motivation is that a boundary with a slightly higher confidence level than the current one would share a similar shape, with closely related points on each boundary. Therefore, given a sufficient number of steps, a continuation process could allow an initial polygon approximation to adapt its shape over time to the desired boundary's shape. This approach is reasonable if a suitable method for continuing boundary polygons is found. For example, we tried using search directions normal to the current polygon at each point, or using $\hat{\psi}$ or the polygon pole of inaccessibility as an internal point to search from towards each current point. We also tried using internal sampling to find an internal point that made the boundary a star domain while being close to the centre of the boundary's area. If this failed, several local points distributed around the inside of the boundary were used for outward searching.

However, this raised the two issues with the method. If the ellipse approximation was already a reasonable approximation of the ℓ_c boundary and this boundary was close to convex, with $\hat{\psi}$ making the boundary a star domain, then stepping up several confidence boundaries would require significantly more computation than simply targeting the ℓ_c boundary to start with (as in **RadialMLEMethod**, Section 4.1.6). This tends to be the case for sufficiently regular likelihoods, which is evident in the models covered in this section. If the previous requirements were not fulfilled, such as with a very concave ℓ_c boundary, then each of the methods attempted for the continuation of the often quite convex $\ell_{c,\text{lower}}$ boundary did a poor job of exploring and adapting to the boundary areas which prevented $\hat{\psi}$ from making the boundary a star domain.

4.1.6 RadialMLEMethod

The **RadialMLEMethod** was inspired by the simplest version of the **ContinuationMethod** covered in Section 4.1.5. Pseudocode for the **RadialMLEMethod** can be found in Appendix E.1 and E.1.3. At some (sufficiently low) confidence level, the ellipse approximation found by **AnalyticalEllipseMethod** will be a good local approximation of the confidence boundary of the bivariate profile log-likelihood function. Therefore, we can use points placed on the ellipse at the desired confidence level as directions to search in from the MLE point, $\hat{\psi}$. Additionally, the points may be very close to the boundary and can decrease the region to search in, as implemented for univariate confidence intervals (Section 2.3.3.1).

This method is the first covered here (with pseudocode) to not explicitly sample the interest parameter space, instead using prior knowledge of the space, allowing it to have the best performance for

sufficiently regular bivariate profiles. Note: `RadialRandomMethod` will do similarly, minus the use of ellipse points to decrease the search brackets, if `num_radial_directions` = `totalPoints` and the MLE point is used as the first internal point. As in `RadialRandomMethod`, this method will not infinite loop if the interest parameter bounds are fully contained within the boundary. This is because when a parameter bound is in the way of reaching the boundary in a given search direction, the point on that bound will be returned as the boundary point. Therefore, this method dramatically improves on all of `Fix1Axis`'s issues.

This method is best suited for convex boundaries or boundaries where the MLE point makes the boundary a star domain. In these cases, the polygon representation of the boundary found is expected to be very good. If this is not the case, then the method will not be able to find areas of the boundary that are ‘hidden’ from the MLE point by other parts of the boundary, and the polygon approximations coverage of the boundary will suffer accordingly. These two cases can be seen in Figure 18 with six boundary points. In this figure, we show an ellipse created from a lower confidence level to most clearly show the search directions from the MLE point; in practice, we form the ellipse at the same confidence level as the desired boundary. Additionally, the ellipse approximation may give little or no information if the model is not well-identified. In this case, switching to `RadialRandomMethod`, e.g. with `num_radial_directions` = `totalPoints` and $\hat{\psi}$ used as the first internal point, would be recommended.

4.1.7 IterativeBoundaryMethod

The `IterativeBoundaryMethod` takes the lessons learned from the previous methods and attempts to create the best combination of all of them. Pseudocode for the `IterativeBoundaryMethod` can be found in Appendix E.1.4. `RadialRandomMethod` and `RadialMLEMethod` are the best of the current methods in addressing the issues with `Fix1Axis`, but they still have their issues. Namely, sampling internal points within `RadialRandomMethod` may be inefficient. Likewise, the boundary found by `RadialMLEMethod` may not sufficiently represent the bivariate boundary if $\hat{\psi}$ does not make the boundary a star domain. Therefore, the `IterativeBoundaryMethod` seeks to utilise the good initial coverage found by these two prior methods, with the ability to iteratively search for regions where it can improve on the current boundary approximation. Note: we use the ITP method [115] as the bracketing method when searching for new boundary points instead of Brent’s method as we observed a > 2 times slowdown in this specific case, which is likely due to implementation details. Changing the implementation to make Brent’s method work appropriately was attempted but featured undetectable issues.

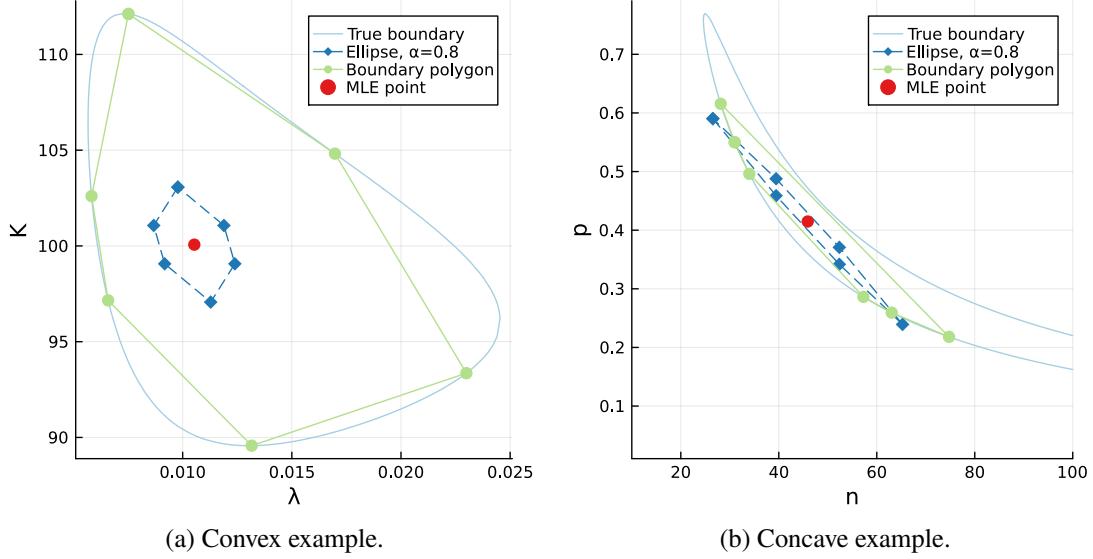


Figure 18: Examples of using RadialMLEMethod on convex and concave bivariate boundaries. The locations on the ellipse relative to the MLE point give the search directions for each location towards points on the boundary polygon. The convex boundary is from the first two parameters of the logistic model [1]. The concave boundary is from the two parameters in the normal approximation of a binomial distribution (Appendix A.4). The start point shift is set to 0.0, and equal spacing on the ellipse is used.

This method starts by finding an initial boundary approximation by pushing out from the MLE point in directions defined by either RadialMLEMethod or RadialRandomMethod (i.e. starting from their solutions where $\hat{\psi}$ is the only internal point used). The method then seeks to improve this boundary by minimising an internal angle and an edge length objective; each considered sequentially until the desired number of boundary points is found. As such, the method can be thought of as a mesh smoothing or improvement algorithm; we can consider the boundary found at a given moment in time to be an N-sided polygon with edges between adjacent boundary points (vertices).

Regions we define as needing improvement are those with:

1. Internal angles between adjacent edges that are far from being straight (180 degrees or π radians). The region defined by these edges is not well explored, as the real boundary edge in this region likely has some degree of smooth curvature. It may also indicate that one of these edges cuts our boundary into an enclosed region and an unexplored region on the other side of the edge. If a vertex is on a user-provided bound for a parameter, this objective is set to zero, as this angle region is a byproduct of user input and not the actual log-likelihood region.
2. Edges between adjacent vertices that have a large Euclidean distance. The regions between these vertices are not well explored as it is unknown whether the boundary is straight between them. On average, the closer two vertices are, the more likely the edge between the two points

is well approximated by a straight line.

In order to improve these regions, we iteratively consider sets of candidate edges that currently maximise each objective and explore the region around these edges. Objectives are rescaled by the relative magnitude of interest parameters so that the weighting of each parameter is approximately equal. To explore new regions, we place a candidate point, ψ_{guess} , at the midpoint of a candidate edge. If ψ_{guess} is on the boundary, we accept the point. If ψ_{guess} is inside the boundary, we search in the normal direction to the edge until we find a boundary point or hit the parameter bound, accepting either. If ψ_{guess} is outside the boundary, we find the edge of the boundary polygon opposite to our candidate edge (in the normal direction). We then search between the closest vertex on the opposite edge, subject to the vertex not being on the candidate edge. This guarantees that our search direction will hit a boundary point (see Appendix F.1 for further discussion).

As in `RadialRandomMethod`, we distort the search direction by the relative magnitude of the two interest parameters. However, unlike `RadialRandomMethod`, we need to use the square of the relative magnitude, as discussed in Appendix F.2. Using a log transformation to deal with different parameter scales in this method may be more appropriate as no sampling is used. However, this was not implemented directly for consistency with the other methods. Additionally, this method can still use a log transformation by directly transforming the supplied log-likelihood function.

An example of this method stepping through three iterations on a convex boundary is shown in Figure 19. Appendix F.3 shows an equivalent example on a concave boundary.

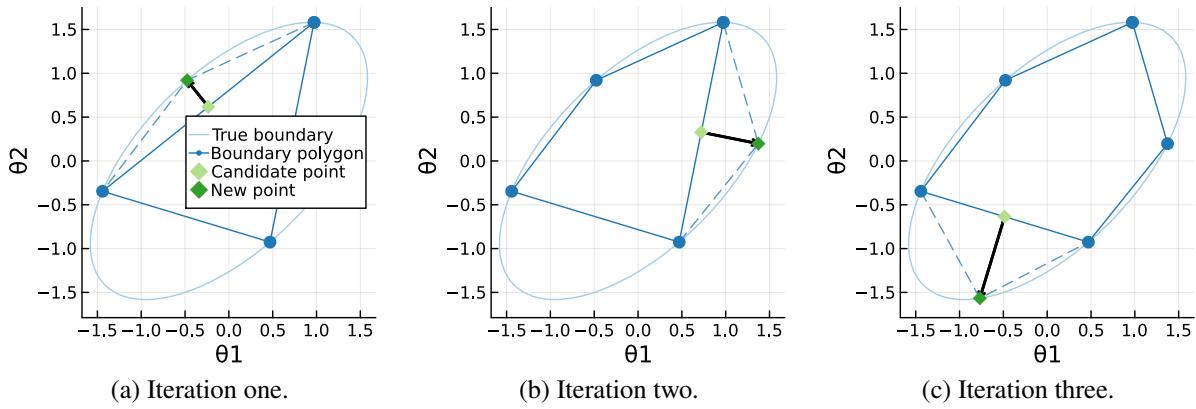


Figure 19: Three iterations of `IterativeBoundaryMethod` starting from a three-point boundary found using `RadialMLEMethod`. The start point shift is set to 0.15, and equal spacing on the ellipse is used. The first and third iterations consider the angle objective, and the second iteration considers the edge length objective. Dashed blue lines show the new edges of the polygon. Parameter magnitudes are both one.

In Figure 19, the 95% confidence log-likelihood boundary is described by an ellipse with major and

minor axis radii of two and one, respectively, rotated by 0.25π radians and centred at $[0, 0]$. Here, `IterativeBoundaryMethod` starts from the `RadialMLEMethod` solution with an initial three points, with no clustering, and a start point shift of 0.15. In iteration one (Figure 19a), the vertex with the largest angle objective is the top vertex, while the vertex with the second largest is the leftmost vertex. Therefore, the candidate edge is between the top and leftmost vertices. In iteration two (Figure 19b), the longest edge is between the top and bottom right vertices. Therefore, it is the candidate edge. In the third iteration (Figure 19c), the vertex with the largest angle objective is the leftmost vertex. The adjacent vertex with the largest angle objective is the bottom vertex (anti-clockwise from the leftmost vertex). Therefore, the bottom left edge is the third candidate edge. After only three iterations (and six total points), the polygon approximation rapidly converges to a good approximation of the bivariate boundary, indicating that the method is effective as a polygon improvement heuristic.

4.2 Sampling Inside Profile Boundaries

To sample inside a polygon approximation of the bivariate boundary, we first create a polygon hull of the found points. We can use either a minimum perimeter polygon (MPP), a convex hull or a concave hull method to form the hull, where each has advantages and disadvantages. These advantages and disadvantages centre around the sampling efficiency of points within each hull and the incorporation of known internal points, which may improve the quality of the representation.

The MPP approach will generally be the best for sampling efficiency and representation quality if the points found are well spread around the boundary. Notably, convex and concave hull methods can incorporate known internal points to create a polygon hull, improving their representation relative to MPP, particularly for sampling-based boundary methods. The main disadvantage of using a concave hull method is that it is a heuristic which may ignore good boundary points when producing the hull, worsening the representation. Similarly, the main disadvantage of the convex hull approach is that the rejection rate of sampling within the convex hull may become very high for concave boundaries. This can occur if the area of the convex hull of the bivariate boundary is much larger than that of the bivariate boundary. These advantages and disadvantages are further discussed in `LikelihoodBasedProfileWiseAnalysis`' documentation [3].

Once we have this polygon hull, we can sample within this polygon using a rejection approach (uniform random, grid-based or Latin Hypercube sampling methods). The uniform random approach can be used to reach the desired number of points exactly without any guarantees on point distribution. In contrast, the grid-based and Latin Hypercube methods give a greater guarantee of the even distribution of points while not guaranteeing the desired number. These methods create a square bounding box

around the hull, scaling the number of points to sample by the relative areas of the box and the hull. Then, once the points are sampled in interest parameter space, they undergo a first rejection step by testing if they are contained in the hull. Surviving points then undergo a second rejection step by testing if the profile log-likelihood function at that point is greater than or equal to the confidence threshold, ℓ_c . If they survive the second rejection step as well, then they can be added to the bivariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^0)$.

4.3 Performance of Methods on Logistic Model

Here, we will consider the performance of all the methods covered so far aside from `AnalyticalEllipseMethod` and `ContinuationMethod` on the three-parameter logistic model [1] (Appendix A.1). In particular, we will consider the number of optimisation function calls required to find increasing numbers of boundary points on the likelihood-based confidence sets for bivariate parameters at a 95% confidence level. We will then consider how well the polygon hull formed by the minimum perimeter polygon of these boundary points approximates the bivariate boundary; to test this coverage, we follow the procedure discussed in Section 2.3.8. This second consideration is significant for allowing our parameter confidence sets to have the correct coverage level. These two considerations will allow us to determine the method and relevant settings for the experiments in subsequent sections. We will then compare this chosen method's optimisation function call performance against rejection sampling approaches.

Likelihood function calls are not strictly required for comparing between methods. For large numbers of optimisation function calls, we can expect the relationship between the two to scale linearly (a corresponding plot will be placed in an appendix to demonstrate this). However, they will be useful in later sections when comparing our chosen method against other methodologies. Additionally, the cost of finding the hessian at the MLE point is non-zero but will be ignored in our comparisons here. As noted for univariate confidence intervals in Section 3.1.3.5, there are up to three duplicate optimisation function calls per boundary point that we can eliminate from the following graphs by implementing a bespoke version of Brent's method.

The true and analytical ellipse approximation of the 95% confidence bivariate boundary for each interest parameter pair under a particular synthetic data realisation is seen in Figure 20. This example is good for demonstrating the weaknesses of `Fix1Axis` and `SimultaneousMethod`, with the second and third boundaries featuring points located very close to the lower bound of 0 on the parameter $C(0)$. As a result, points in this location will be tricky for these methods to find. It also demonstrates some of the weaknesses in `RadialMLEMethod`. The first and third boundaries are convex and will be

straightforward to find, while the second boundary is concave, which may prevent this method from finding the right-most part of the boundary.

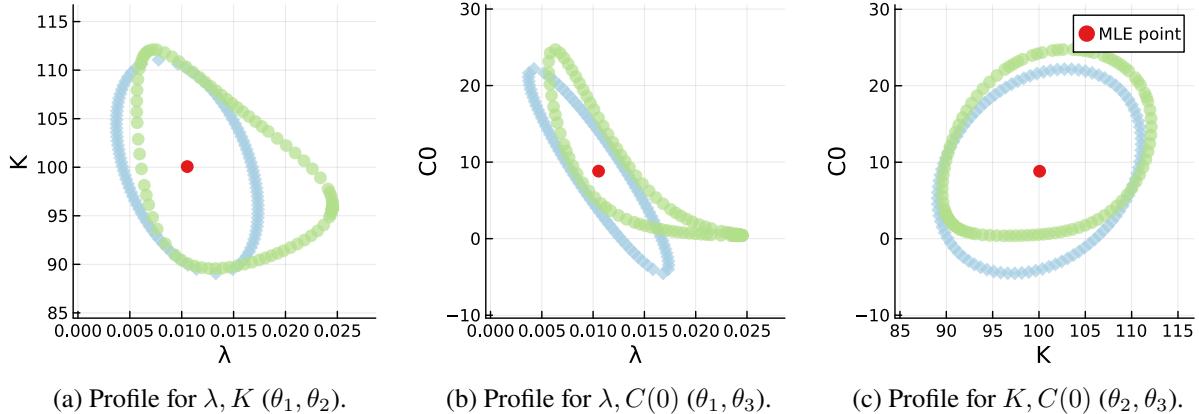


Figure 20: Points located on the true 95% confidence bivariate boundary (green dots) and ellipse approximation of the boundary (light blue diamonds) for each parameter pair in the logistic model [1]. The ellipse approximation can go non-feasible (have values of $C(0) < 0$).

The settings used by each method can significantly impact performance in terms of boundary coverage, and the number of optimisation function calls. To mitigate this, we use relatively generous settings comparable across relevant methods. The MLE point is always used as an internal point (when supported), and the parameter values of `RadialMLEMethod`, where relevant, match those of `IterativeBoundaryMethod`. The minimum proportion of unique internal and external points in `SimultaneousMethod` is set to 0.5 to balance the method’s boundary coverage and number of function evaluations. A second instance of `IterativeBoundaryMethod` is included to demonstrate that it is similarly effective when starting from `RadialRandomMethod`. The full details of these settings are in Table 5.

Method	Settings
<code>Fix1AxisMethod</code>	NA
<code>SimultaneousMethod</code>	Minimum unique proportion = 0.5, use MLE point
<code>RadialRandomMethod</code>	Number of radial directions = 5, start point shift $\sim \text{rand}(0, 1)$ per internal point, use MLE point
<code>RadialMLEMethod</code>	Start point shift = 0.15, clustering parameter = 0.1
<code>IterativeBoundaryMethod</code>	Start from <code>RadialMLEMethod</code> with 10 points, start point shift = 0.15, clustering parameter = 0.1; five angle points and five edge points per iteration
<code>IterativeBoundaryMethod</code>	Start from <code>RadialRandomMethod</code> with 10 points (radial directions= 10, use MLE point, start point shift = 0.15); five angle points and five edge points per iteration

Table 5: Settings used for performance comparisons between bivariate methods.

4.3.1 Optimisation Function Call Performance

The mean optimisation call performance of each method for increasing numbers of boundary points on the three bivariate 95% confidence sets of the logistic model is shown in Figure 21. The corresponding mean likelihood call performance, and the optimisation call performance for each bivariate profile can be found in Appendix G.1.1. To demonstrate the impact of widening parameter bounds on methods based on rejection sampling, we run the same example with 50% wider parameter bounds (Figure 21b). Increasing the width of each parameter bound by 50% increases the 2D sample area for rejection methods by 125% ($(1.5 \times 1.5 - 1.0 \times 1.0) / (1.0 \times 1.0) \times 100\% = 125\%$).

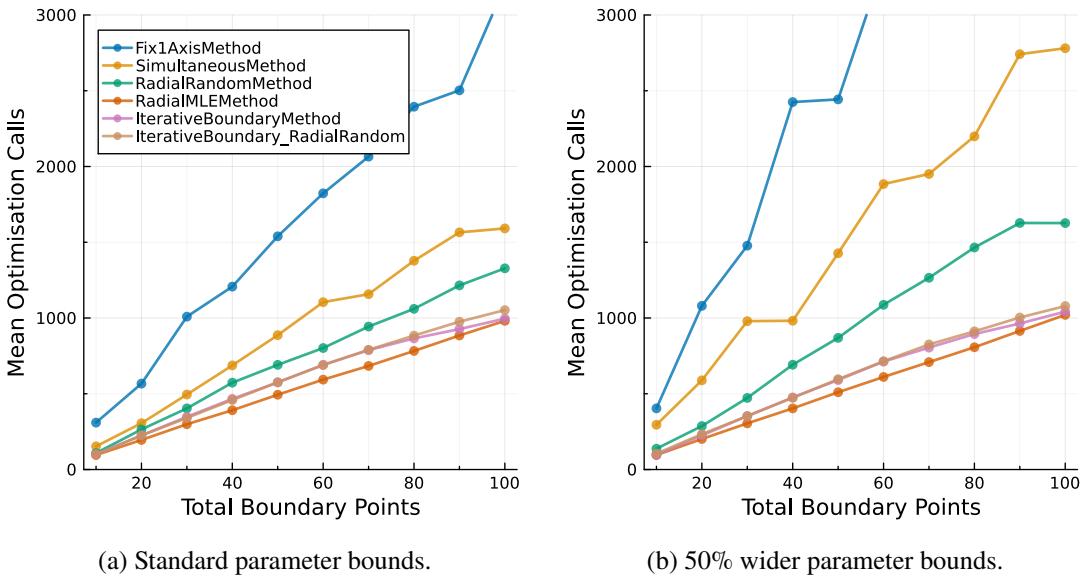


Figure 21: Mean number of optimisation calls required to find a desired number of boundary points on each of the bivariate profiles in the logistic model [1] as seen in Figure 20, for each method. Figure 21b shows the impact of increasing the width of each bound by 50%: from $(a, b) = ([0.0, 50, 0.0], [0.05, 150, 50])$ to $(a, b) = ([0.0, 25, 0.0], [0.075, 175, 75])$.

In Figure 21, the scaling behaviour for each method is approximately linear for increasing numbers of boundary points. Additionally, increasing the width of the parameter bounds by 50% causes a marked increase in the number of optimisation calls required for the same number of boundary points for **Fix1AxisMethod**, **SimultaneousMethod** and **RadialRandomMethod**, due to their reliance on rejection sampling. In contrast, this width change makes no noticeable impact on the number of optimisation calls required by **RadialMLEMethod** and **IterativeBoundaryMethod**.

Fix1AxisMethod is the worst performer, scaling with around two times worse performance than all other methods. It requires around 30 calls per boundary point found with standard parameter bounds and around 60 calls with widened bounds (note: it goes off the graph, which had limits fixed to compare and consider other methods more easily). **SimultaneousMethod** is a significant improvement on

`Fix1AxisMethod` in optimisation call performance. It requires around 16 calls per boundary point (30 with widened bounds) due to greatly improving sampling efficiency by throwing away fewer points. Meanwhile, `RadialRandomMethod` sits between the performance of `SimultaneousMethod` and the best performers. It requires around 14 calls per boundary point (18 with widened bounds), likely due to the fewer internal points it has to sample for each boundary point. Under the current settings, it requires one internal point for every five boundary points, while `SimultaneousMethod` requires one for every two boundary points. The performance of the remaining methods, `RadialMLEMethod` and both `IterativeBoundaryMethod` settings, is functionally identical. They both require around ten calls per boundary point regardless of parameter bounds; this could be reduced by around 30% by implementing a bespoke implementation of Brent's method.

4.3.2 Coverage of True Bivariate 95% Confidence Set Boundary

The approximate area coverage of the polygon found by each method for increasing numbers of boundary points on the ‘true’ bivariate 95% confidence set boundary was considered for 500 iterations. Here, we use the MPP approach to create the polygon hull. The example is replicated with a convex hull approach in Appendix G.1.2, which improves results for sampling-based methods. However, the conclusion remains unchanged. The synthetic observed data was independently generated for each distinct iteration and was consistent across each method and number of boundary points. Between 250 and 400 internal points were found to check polygon area coverage in each iteration. The results of this testing can be found in Figure 22. The 0.05 quantile of observed coverage is shown here because of its high confidence in selecting a method and minimum number of boundary points with good performance. Based on this quantile, we would expect area coverage to reach a specified level 95% of the time given a specified number of points in this example. This expectation will be important for parameter coverage tests and coverage tests of predictive quantities.

As expected, the concave boundary from the second set of parameters (Figure 22b) was more challenging to find for every method, with coverage lower for the same number of boundary points. The sampling-based methods and `RadialMLEMethod` all fail to converge to area coverage higher than 95%; the former due to difficulty sampling inside and outside of the bottom right-hand part of the boundary, and the latter due to the MLE point not being a point that makes the boundary a star domain. The bottom right-hand part cannot be found with `RadialMLEMethod`.

Of the sampling-based methods, `RadialRandomMethod` has the best coverage guarantees on each boundary for the same number of points. This is because it searches radially with spaced-out directions from each internal point, enabling better local exploration. Additionally, it does not have issues finding

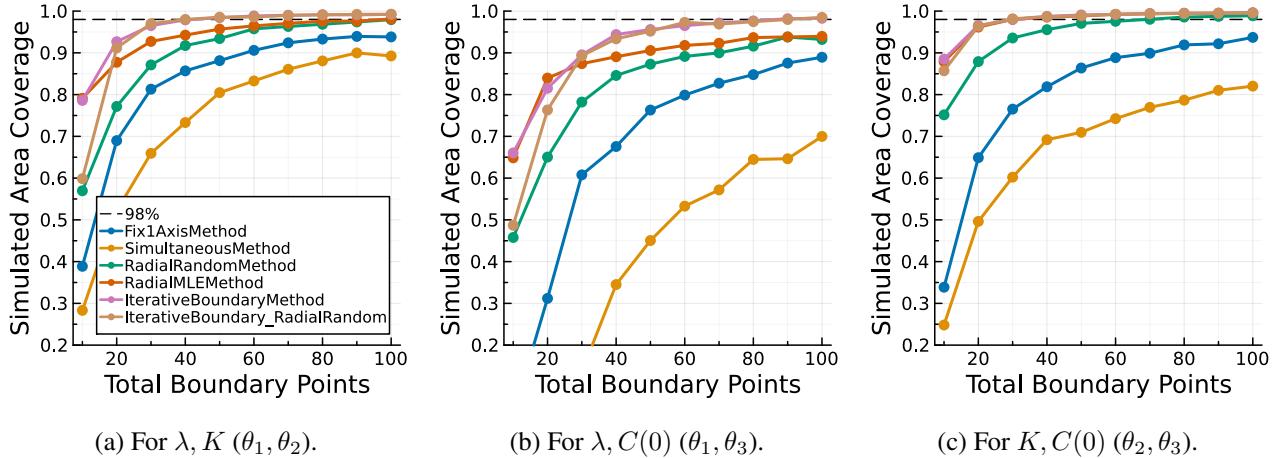


Figure 22: Approximate area coverage of the boundary polygon found from each method for increasing numbers of boundary points on each of the bivariate profiles in the logistic model [1], as seen in Figure 20.

boundaries near bounds due to not needing to sample a point between the boundary and bound. The other sampling methods perform worst, with the **SimultaneousMethod** having much worse coverage than **Fix1AxisMethod** at the same number of boundary points. This is likely due to having poorer search directions or less well-informed point pairs than the more constrained **Fix1AxisMethod** approach.

IterativeBoundaryMethod can converge to more than 98% coverage on every boundary. In contrast, **RadialMLEMethod** requires more points to do so on the first and third boundaries and gets stuck at 94% coverage on the second boundary. A similar phenomenon, but with worse performance, can be observed for **RadialRandomMethod**.

Starting **IterativeBoundaryMethod** from **RadialRandomMethod** initially shows worse coverage than starting with **RadialMLEMethod** on the first two boundaries (Figures 22a and 22b). However, this recovers to effectively the same level for 20 or more points. This is because the ellipse approximation used to choose radial directions by **RadialMLEMethod** contains more information about these boundaries. Similarly, **RadialMLEMethod** is excellent for the highly elliptical third boundary, matching the coverage of **IterativeBoundaryMethod** for a given number of points, but cannot match it on the other more concave boundaries.

Therefore, **IterativeBoundaryMethod** is the best method for area coverage, reaching at least 98% coverage for fewer total boundary points than all other methods. Additionally, we found it had the best optimisation call performance in Section 4.3.1. It will be used in all subsequent experiments, starting from the **RadialMLEMethod** solution (as this setting displays slightly better area coverage). In experiments involving the logistic model, we will use 50 boundary points, allowing the method to

have very high boundary area coverage at least 95% of the time.

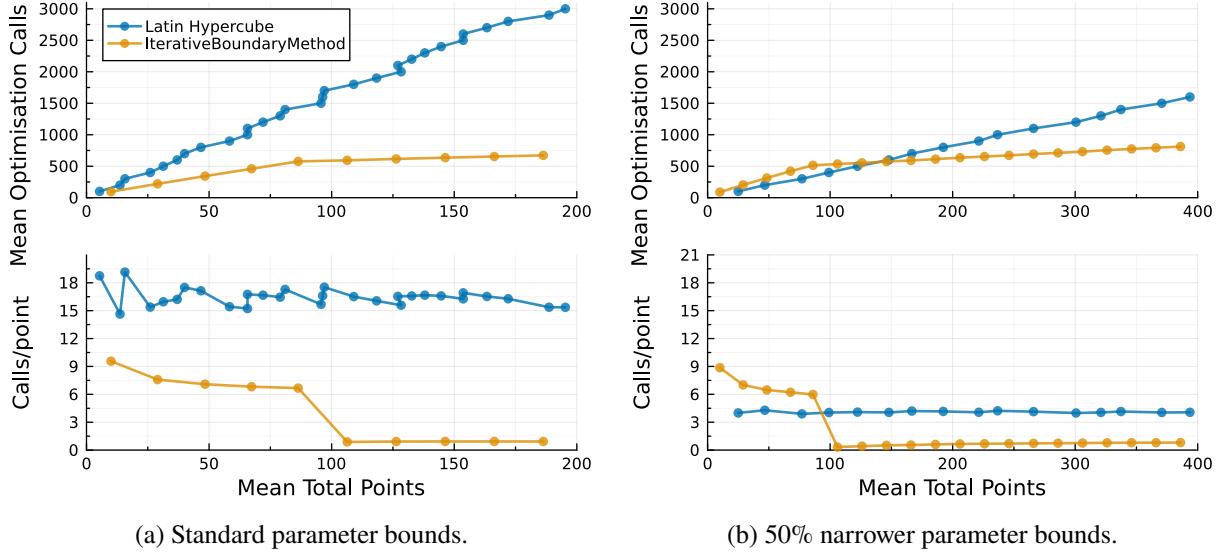
4.3.3 Comparing Optimisation Function Call Performance to Rejection Sampling

To motivate the use of `IterativeBoundaryMethod` when computing bivariate confidence sets, we compare it to a rejection sampling approach. For this comparison, we use the number of optimisation function calls required to find the same number of points. Additionally, we consider how first finding the boundary and then sampling inside that boundary impacts performance. Finally, we consider the impact of reducing the width of interest parameter bounds by 50%. Decreasing the width of each parameter bound by 50% decreases the 2D sample area for the rejection sampling approach by 75% ($(1.0 \times 1.0 - 0.5 \times 0.5) / (1.0 \times 1.0) \times 100\% = 75\%$). These bounds are adjusted to ensure they still contain the boundary shown in Figure 20.

In this example, the `IterativeBoundaryMethod` requires about ten calls per boundary point, regardless of parameter bounds (Section 4.3.1). This statistic ignores that internal points found while searching for boundary points are kept, which improves the call per (any) point performance. Therefore, for a rejection sampling method to exhibit better performance when finding points inside the confidence set, the area of the set in interest parameter space would need to be greater than at least 10% of the area of the interest parameter bounds. Even if this was the case, the points found using rejection sampling would potentially be lower value points for propagating forward uncertainty than those on the confidence boundary. As a result, a rejection sampling method may need to be significantly more computationally efficient than our method to offset the difference in the value of the points found. The value of these internal points relative to boundary points is explored in Section 5.3.

In Figure 23, we compare the number of optimisation function calls used by a Latin Hypercube rejection sampling approach and the `IterativeBoundaryMethod` with the number of bivariate confidence set points found. Once the `IterativeBoundaryMethod` reaches 50 boundary points, we use a Latin Hypercube rejection sampling approach to sample points within this boundary, as covered in Section 4.2. This can be seen as the gradient change after 80 total points (boundary plus internal points); finding new confidence set points is very efficient using the sampling approach within the boundary polygon.

With the original parameter bounds (Figure 23a), the rejection sampling approach is significantly worse than the boundary method approach, even before the gradient change. When the `IterativeBoundaryMethod` reaches ≈ 90 points, it has used around 600 optimisation calls, whereas the rejection sampling approach has used around 1500 or 2.5 times more. This difference becomes even more significant for double the points (≈ 180). In this case, the boundary method coupled with sam-



(a) Standard parameter bounds.

(b) 50% narrower parameter bounds.

Figure 23: The mean number of optimisation calls required to find a mean number of bivariate confidence set points across each bivariate profile in the logistic model [1]. For a Latin Hypercube rejection sampling approach and the `IterativeBoundaryMethod` in conjunction with a Latin Hypercube method after 50 boundary points are found. The calls per point metric takes into account the gradient change for the `IterativeBoundaryMethod` after around 80 total points. Figure 23b shows the impact of decreasing the width of each bound by 50%: from (a, b) = ([0.0, 50, 0.0], [0.05, 150, 50]) to (a, b) = ([0.005, 75, 0.0], [0.0300, 125, 25]).

pling inside the boundary takes around 700 optimisation calls, while the rejection sampling approach uses around 2800 or four times more.

This performance difference is significantly reduced when narrowing the parameter bounds by 50%. However, narrowing the bounds requires effectively perfect knowledge of the location of the profile. The interest parameter bounds we use here are very close to being the minimum and maximum point values of the bivariate profiles in Figure 20. We see that for less than 150 mean total points, the rejection sampling approach used fewer mean optimisation calls, at around four calls per point. This is significantly improved from the around seven calls per point required by the `IterativeBoundaryMethod` ($600/90 \approx 7$). However, once the `IterativeBoundaryMethod` switches to sampling points inside the boundary, it undercuts the performance of the rejection sampling approach for more than 150 mean total points. This is because sampling points inside the boundary cost just over one optimisation call per point.

This example shows that the rejection sampling approach can be more performant than our method, but it requires excellent knowledge of the profile location. Additionally, it will get overtaken by our method for large numbers of points irrespective of location knowledge. Therefore, our methodology is more performant and appropriate for sampling points from a bivariate profile with arbitrary parameter

bounds. This performance gap would be even more significant if `find_zero` was replaced with a bespoke implementation of Brent’s method, eliminating the three duplicate optimisation calls per boundary point. This demonstrates that rejection sampling approaches should not be used for finding unimodal bivariate profiles over our methodology unless they are first paired with a boundary polygon, as used here.

4.4 Coverage of True Bivariate Boundary Additional Models

Here, we consider the approximate area coverage of the boundary polygon created by `IterativeBoundaryMethod` on the Lotka-Volterra [1] and two-species logistic with logit-normal data distribution [2] models (Appendix A.2 and A.3.1). Here, we use the MPP approach to create the polygon hull. This is seen in Figure 24. We consider this over 100 iterations on both examples. In the Lotka-Volterra and two-species examples, around 200 to 400 and 100 to 200 internal points were found to check polygon area coverage in each iteration. The bivariate profiles from these models are visualised in Appendix H.

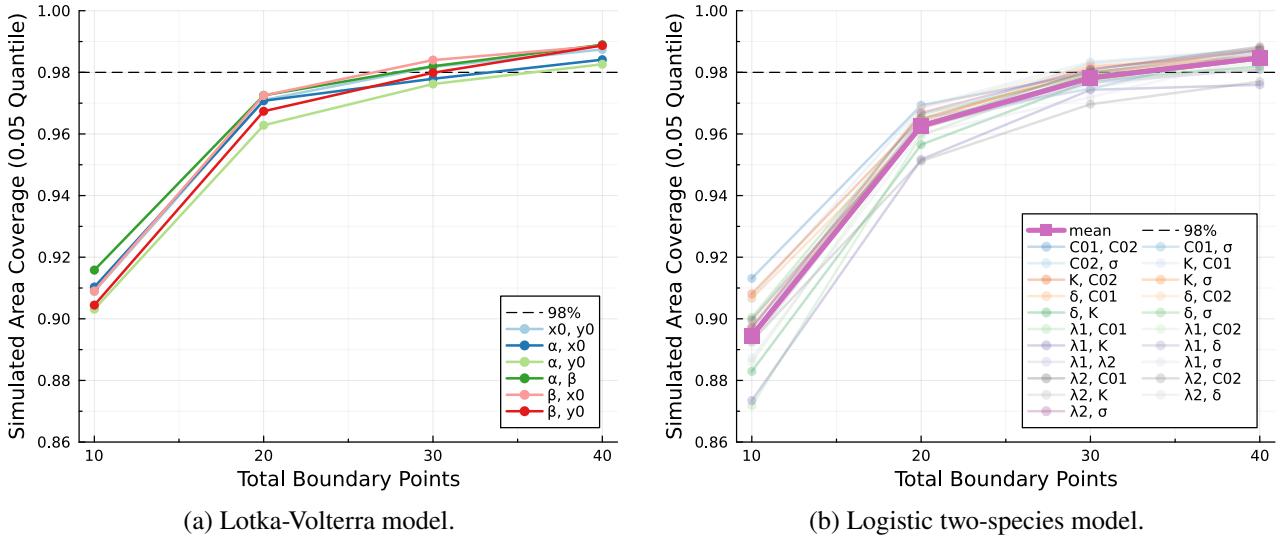


Figure 24: Approximate area coverage of the boundary polygon found from the `IterativeBoundaryMethod` for increasing numbers of boundary points on each of the bivariate profiles in the Lotka-Volterra [1] and logistic two-species [2] models.

The bivariate profiles in the Lotka-Volterra model are very well approximated by the ellipse approximation (Appendix H.1), so we started the `IterativeBoundaryMethod` from `RadialMLEMethod` with 20 points for 20 or more total boundary points. In Figure 24a, only 30 boundary points are required to reach about 98% area coverage in every bivariate profile, at least 95% of the time. Therefore, we will use this version of the method with 30 boundary points for subsequent experiments on this example.

The bivariate profiles in the two-species logistic model were also well approximated by the ellipse approximation (Appendix H.2). Note: the profiles shown in that appendix are larger than those found in the coverage example here, as they were produced from the real data; the real data has a higher expected data distribution standard deviation than the standard deviation used here (around 0.45 rather than 0.1). Resultantly, the ellipse approximations are even more consistent with the boundary profiles. As a result, we switch to the `RadialMLEMethod` for this example (i.e. `IterativeBoundaryMethod` starting from `RadialMLEMethod` with the total number of boundary points). Additionally, there is a more significant difference in parameter magnitude in this example than in previous examples, so we reduce the clustering parameter by an order of magnitude to 0.01. As in the Lotka-Volterra model, in Figure 24b only around 30 boundary points are required to reach an average of 98% area coverage, at least 95% of the time, across all 21 bivariate profiles. As a result, we will use this version of the method with 30 boundary points for subsequent experiments on this example.

4.5 Bivariate Parameter Coverage of Profile Boundaries

This section will demonstrate the frequentist coverage properties of the boundaries of the profile likelihood-based bivariate confidence sets in the same fashion as the confidence interval coverage evaluation in Section 3.2. We follow the procedure discussed in Sections 2.3.7 and 2.3.7.2 to test this coverage. Again, the expected coverage properties are already shown in literature [24]; we demonstrate it here to validate our implementation and the appropriateness of the PWA workflow on the models considered. We will focus on the coverage of bivariate 95% confidence boundaries. The settings used in `IterativeBoundaryMethod` to find the profile boundary for each example can be found in Table 6.

Model	Total Boundary Points	Settings
Logistic	50	Start from <code>RadialMLEMethod</code> with 10 points, start point shift = 0.15, clustering parameter = 0.1; five angle points and five edge points per iteration
Lotka-Volterra	30	Start from <code>RadialMLEMethod</code> with 20 points, start point shift = 0.15, clustering parameter = 0.1; five angle points and five edge points per iteration
Logistic Two-Species	30	Start from <code>RadialMLEMethod</code> with 30 points, start point shift = 0.15, clustering parameter = 0.01

Table 6: Settings used for `IterativeBoundaryMethod` on each model.

4.5.1 Logistic and Lotka-Volterra Models

As in the confidence interval coverage evaluation (Section 3.2), the bivariate confidence profile boundaries for both the logistic and Lotka-Volterra models show excellent coverage properties, with all parameters having simulated coverage within 0.02 of 0.95 in Figure 25. The 95% confidence interval for the coverage value given by all but one of the error bars contains 0.95. However, the upper endpoint for the coverage confidence interval for $\beta, y(0)$ is almost at 0.95 (0.948).

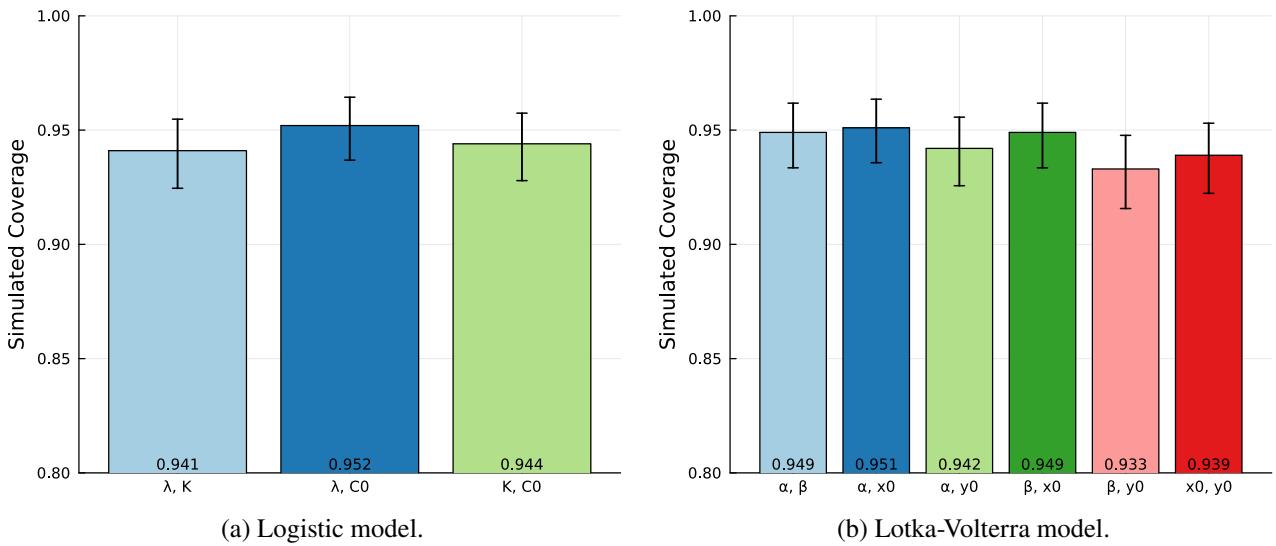


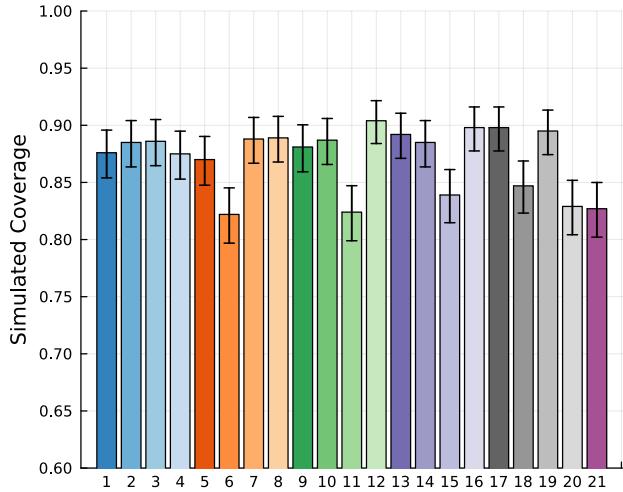
Figure 25: Coverage of bivariate 95% confidence boundaries for each pair of parameters in the logistic and Lotka-Volterra model [1] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

4.5.2 Two-Species Logistic With Logit-Normal Data Distribution

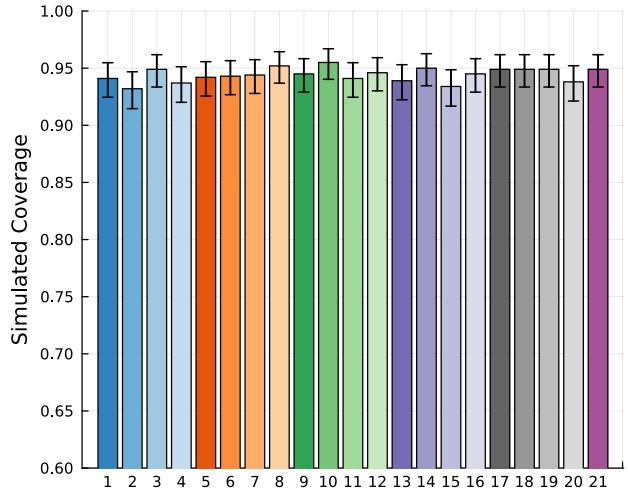
Similarly to the confidence interval coverage evaluation for the two-species logistic model with logit-normal data distribution (Section 3.2.2.1), the simulated coverage of bivariate confidence profile boundaries is lower than the desired coverage of 0.95 in Figure 26a. Interestingly, when the bivariate profile only contains a combination of the first six parameters, which had simulated confidence interval coverage of 0.9 in Section 3.2.2.1, these profile boundaries also have coverage of around 0.9. Similarly, when the bivariate profile contains the seventh parameter, σ , the profile's coverage is around 0.83, or around halfway between the confidence interval coverage of all other six parameters and the coverage for σ at 0.79. This indicates that coverage in the confidence interval space is directly correlated with the coverage of the interest parameter pair in the bivariate profile boundary space.

When we increase the number of observations significantly in Figure 26b, the coverage for all bivariate profiles improves to approximately 0.95. This was also observed in Section 3.2.2.1 for the coverage of parameter confidence intervals. 19 of the simulated coverage 95% confidence intervals now

contain 0.95. Moreover, the remaining two confidence intervals are very close to containing 0.95 (upper endpoints of 0.946 and 0.948 for parameter pairs 2 and 16, respectively). This again shows that the low coverage is due to too few model observations at the asymptotic confidence threshold, $\ell_c = -\Delta_{2,0.95}/2$.



(a) Two-species logistic, logit-normal (22 observations).



(b) Two-species logistic, logit-normal (222 observations).

Figure 26: Coverage of bivariate 95% confidence boundaries for each pair of parameters in the two-species logistic model with logit-normal data distribution [2] from 1000 coverage simulations. Parameter pairs numbers are in order of appearance in Figure 67, Appendix H.2. 95% confidence intervals for the simulated coverage are provided as error bars.

5 Profile-Wise Analysis: Coverage of Predictive Quantities on Simple Models

This section provides a robust quantitative analysis of the coverage properties of profile-wise sets for predictive quantities. This will allow us to better understand the properties of these profile-wise sets in the PWA workflow. Similarly, it will enable us to make, test and implement recommendations to improve its performance. We explore profile-wise predictions from the unions of univariate and bivariate profiles. For comparison, we first analyse the likelihood-based prediction sets formed by propagating forward full parameter confidence sets. The coverage of sets for predictive quantities is investigated using the procedures discussed in Sections 2.3.7, 2.3.7.3 and 2.3.7.4.

We note that the coverage of profile-wise trajectory confidence sets has been explored in other work already [9]; here, we consider it on three of the main models used to demonstrate the use of the PWA workflow [1, 2]. In these particular models, the properties of profile-wise prediction sets have only been visually compared to the sets formed by propagating forward full parameter confidence sets. Descriptions of these models can be found in Appendix A. Additionally, we also introduce the first use of reference tolerance sets in the PWA workflow.

We will consider the computational performance of the profile-wise prediction sets against the prediction sets from the full parameter confidence sets, which remains an important consideration. This has not yet been quantitatively explored in the PWA workflow. However, we delay this discussion until the following section, in which the profile-wise prediction sets have better coverage properties. For the same reason, we also delay considering using subsets of univariate or bivariate profiles to form profile-wise prediction sets with similar coverage performance to the union until the following section. However, we will note when profiles of specific parameters may be more or less important for making specific predictions and begin a discussion on why.

We could also compare the computational and statistical performance of the PWA workflow when making predictions to other methods like the pointwise PPL approach. However, we have chosen against doing this to narrow the scope of the thesis and make our comparison more directly related to the works from which our models are taken. Nonetheless, this remains a valuable opportunity for future work to cement the value of the PWA workflow. For example, similar to work done for parameter identifiability with the profile log-likelihood approach [52] or uncertainty quantification of predictions [45].

We will consider coverage in the logistic and two-species logistic models by looking at a much larger number of timepoints (201) set between the minimum and maximum timepoints of the original observations. On the Lotka-Volterra model, we will again consider a larger number of timepoints

(201) while including timepoints later than those of the original observations (for $t \in [0, 10]$ rather than only $t \in [0, 7]$). We will look at coverage at a 95% confidence level for all models, with 95% population reference intervals used in the observation sections. Our alternative observation coverage goal is to capture 95% of observations at a 95% confidence level. Given the statistical properties of the PWA workflow, we would also expect similar results at different confidence levels and reference interval proportions. Additionally, when we observe poor simultaneous coverage, we will use the simulated pointwise coverage metrics to understand why this is the case [9].

The data distribution standard deviation in the logistic and Lotka-Volterra models is known, so all conclusions found for the trajectory confidence sets will translate 1:1 into the reference tolerance set space. We will still explore the alternative observation coverage goal introduced in Section 2.3.6.2 on these models.

The settings used in this section are those determined in the previous sections:

- We set the absolute tolerance in `find_zero` to $\log_{10}(\text{atol}) = -3$.
- We set the tolerance setting of the numerical optimisation algorithm to $\log_{10}(\text{xtol_rel}) = -12$.
- We use the ‘replace either’ approach for parameter confidence intervals. This approach uses the Wald confidence interval to update the bounds used to bracket each side of the relevant parameter confidence interval.
- We use the settings in Table 6 to find bivariate profiles on each model. We use the Latin Hypercube sampling approach (Section 4.2) to sample points within the bivariate boundary.
- Coverage point estimates and confidence intervals are given to 2 decimal places.

5.1 Full Parameter Vector Likelihood-Based Confidence Sets

We first consider the trajectory confidence set, reference tolerance set and observation coverage results from propagating forward full parameter confidence sets. These sets are in contrast to their approximation formed using the PWA workflow; the profile-wise prediction sets from propagating forward univariate and bivariate profiles. The evaluated reference tolerance sets are used for predicting both the $1 - \delta$ population reference set and individual observations. We use these coverage results as the baseline for comparing coverage and computational expense.

In the three parameter logistic and four parameter Lotka-Volterra models, 10,000 uniform random samples of the full parameter space were used to create the ‘gold-standard’ trajectory confidence set for comparison against the profile-wise prediction sets [1]. We found that the original parameter

bounds used for the Lotka-Volterra model are overly constrained. Resultantly, we will present both the 10,000 sample result from the original bounds and a series of results for the new, wider bounds (Appendix A.2). The ‘gold-standard’ trajectory confidence set was not formed for the two-species logistic model we are considering [2], and we have modified the data distribution model, so our comparison begins from a blank slate on this model.

We evaluate the prediction sets at 201 equally-spaced timepoints within the range of the original data for the logistic and two-species logistic models and between 0 and 10 in the Lotka-Volterra model (rather than only 0 and 7). We observe the population density $C(t)$ in the logistic model. In the Lotka-Volterra model, we observe the number of the prey species $x(t)$ and the predator species $y(t)$. In the two-species logistic model, we observe the population density of the two different coral categorisations $C_1(t)$ and $C_2(t)$. Each sample corresponds to a single evaluation of the likelihood function.

5.1.1 Full Trajectory Confidence Set Coverage

In Figure 27, the full trajectory confidence sets reach simulated coverage of around 0.95 after 30,000 and 500,000 sampled points in the three parameter logistic and four parameter Lotka-Volterra models, respectively. Therefore, the desired coverage level of the model trajectory can be reached on these models when using full trajectory confidence sets. Full trajectory confidence sets formed from 10,000 Latin Hypercube samples, with the original bounds, which should have improved coverage of the parameter space over uniform random samples, have simulated coverage of 0.91 [0.89, 0.92] and 0.86 [0.84, 0.88], respectively. The coverage in both models is relatively acceptable, although lower than desired. This indicates that 10,000 samples are slightly lower than the ideal number for the gold-standard comparisons in [1]. Still, the pointwise coverage of around 0.98 (Figure 27c) indicates that the full trajectory confidence set from 10,000 samples at the original bounds in the Lotka-Volterra model will not be too much smaller, on average, than sets from larger numbers of samples.

The full trajectory confidence set in the seven parameter two-species logistic model requires the evaluation of more than ten million samples to reach simultaneous coverage of 0.90 [0.84, 0.93] from 200 coverage simulations (Figure 28). The upper endpoint for the confidence interval is slightly lower than 0.95 (0.93). Therefore, the desired coverage is at least close to being reached on this model when using full trajectory confidence sets. We would expect this to be around the upper bound for coverage that is able to be reached using the PWA workflow, which produces an approximation of these full trajectory confidence sets.

Ten million samples equates to full parameter confidence sets that contain around 1000 points. This

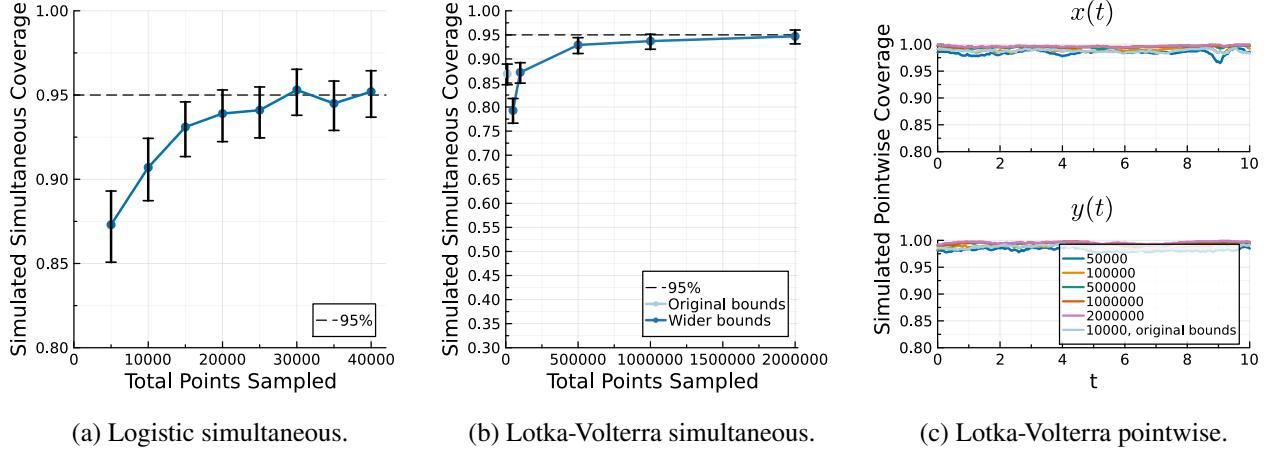


Figure 27: Full trajectory confidence set coverage in the logistic and Lotka-Volterra models [1] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

is with a relatively well-informed set of parameter bounds that may be overly constrained occasionally (Appendix A.3.1). Given the computational power available, these bounds are much tighter than those used for profile evaluation to make sampling of the full parameter confidence set feasible. Resultantly, the number of likelihood calls required to reach the same level of coverage is *much lower* than it would be with the same bounds used for profile evaluation. These parameter bounds were chosen by choosing values slightly wider than simultaneous parameter confidence intervals observed from a single coverage iteration.

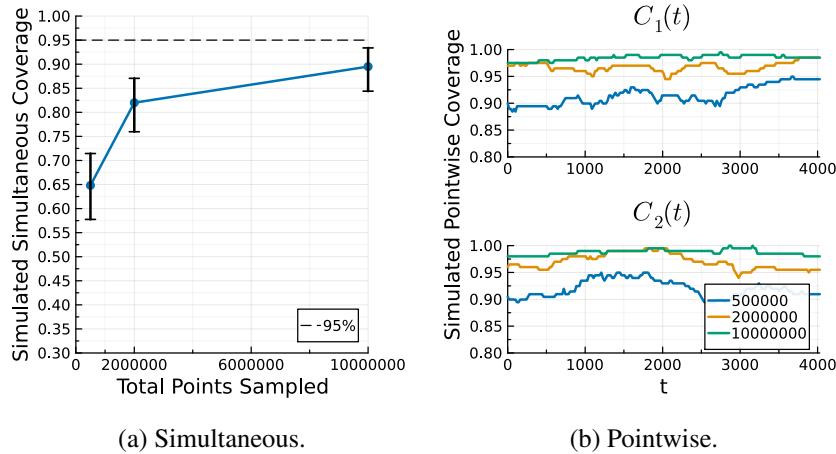


Figure 28: Full trajectory confidence set coverage in the two-species logistic model with logit-normal data distribution [2] from 200 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

The pointwise coverage on this model in Figure 28b hovers between 0.975 and 0.99, in contrast to the pointwise coverage for the Lotka-Volterra model, which was consistently close to 1.0 after one million samples (Figure 27c). This difference in pointwise coverage explains why the simultaneous

coverage reached 0.95 in only one of these models. Sampling a significantly higher number of points to try and reach 95% simultaneous coverage is unrealistic on my machine even with the much tighter parameter bounds; 200 iterations (ideally, we would use 1000) with ten million samples each took around 4 hours, even when multi-threading likelihood evaluations across ten threads.

Such a high number of samples is required because the model has seven parameters, up from the three and four of the previous two examples, demonstrating the curse of dimensionality. This approach will continue to become more infeasible for even higher dimensional models. This is why the PWA workflow approach of profile-wise predictions was developed: to provide an approximation which is significantly less expensive to compute, making sets with frequentist properties more accessible in this problem space[1]. It is also possible that the coverage will not reach 0.95 in this example at the asymptotic confidence threshold, even with a higher number of samples, due to too few observations, as observed for the univariate and bivariate confidence profiles in Sections 3.2.2.1 and 4.5.2, respectively.

5.1.2 Full Reference Tolerance Set Coverage

The full reference tolerance set coverage for the logistic and Lotka-Volterra models is identical to the full trajectory confidence set coverage on these examples in Section 5.1.1 because the data distribution standard deviation is known (see Appendix I).

The full reference tolerance set coverage for the two-species logistic model can be seen in Figure 29. At each number of samples, we observe similar levels in the pointwise coverage of the $1 - \delta$ population reference set in Figure 29b and of the model trajectory in Figure 28b. In contrast, the point estimate for the simultaneous coverage of the $1 - \delta$ population reference set is consistently higher (0.75 vs 0.65, 0.85 vs 0.825 and 0.92 vs 0.90), and the confidence interval at ten million samples now includes 0.95 ([0.87, 0.95]). However, the confidence intervals for the coverage of both prediction sets for two million and ten million point samples each contain the other's point estimate, so the increase may not be statistically significant. Nonetheless, the full reference tolerance set can reach the desired coverage level of the $1 - \delta$ population reference set on this model.

This difference in the simultaneous coverage of the model trajectory and $1 - \delta$ population reference set may be due to differences in the random Latin Hypercube scheme chosen for each iteration across the simulation. It could also be because many standard deviation parameter values in the sampled confidence sets are slightly higher than the true value used to simulate observations. If this is the case, we might expect the full reference tolerance sets coverage to increase relative to the full trajectory confidence sets coverage, and vice versa, if the value was slightly lower than the true value. This is because the width of the $1 - \delta$ reference set evaluated at each full trajectory confidence set point would

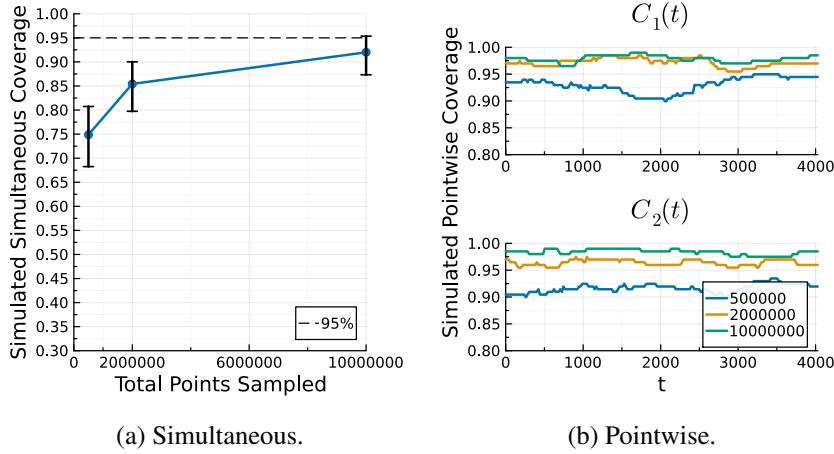


Figure 29: Full reference tolerance set coverage in the two-species logistic model with logit-normal data distribution [2] from 200 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

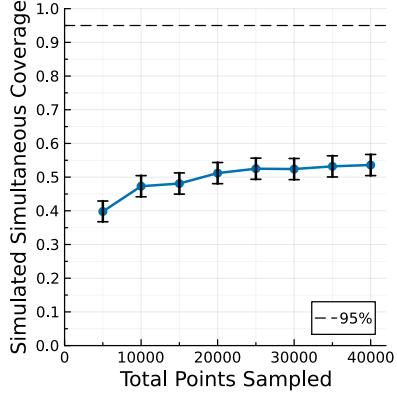
be wider with a higher than true standard deviation, creating reference tolerance intervals with widths larger than those created from the true standard deviation.

However, we do not expect this result to be systematic in the PWA workflow when the data distribution standard deviation parameter is estimated. Further testing in Appendix J indicates that we would expect full trajectory confidence set coverage and full reference tolerance set coverage to be very similar when θ^0 is estimated, although likely slightly different.

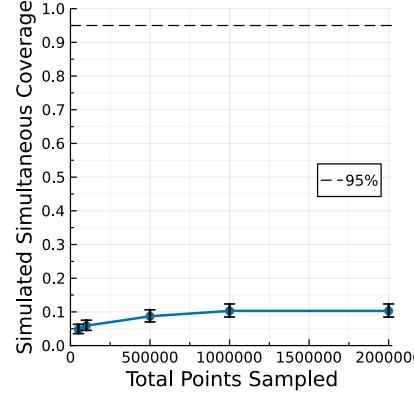
5.1.3 Observation Coverage

When instead considering the more traditional simultaneous coverage of observations and our alternative goal we see that the coverage of observations simultaneously is significantly lower than 0.95 in both the logistic (Figure 30a) and Lotka-Volterra (Figure 30b) models, irrespective of the number of sampled points. This is unsurprising, given that we are using a (95%, 95%) reference tolerance set for a job that technically requires a (100%, 95%) reference tolerance set. In contrast, the alternative observation coverage goal from Section 2.3.6.2 shows higher than 95% coverage (effectively 100% coverage) when the full reference tolerance set's coverage of the 95% population reference set is around 95% (Figures 30d and 30e). This is because the pointwise observation coverage is consistently at least 0.975 for both models and closer to 0.99 for the logistic model for the considered numbers of sampled points (Figures 30g and 30h). This difference in pointwise coverage explains the differences in simultaneous coverage between the two models; in the logistic model, it is close to 0.54 [0.50, 0.56], while in the Lotka-Volterra model, it is barely above 0.0.

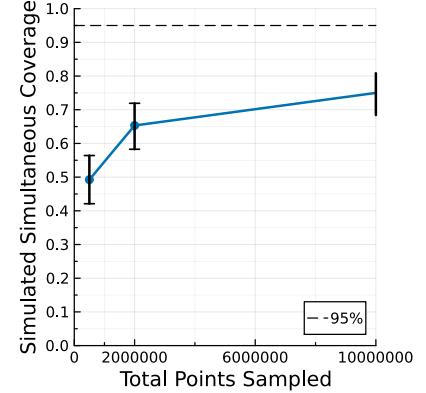
If the pointwise coverage is very high, on a per iteration basis, the proportion of reference tolerance sets that contain at least 95% of testing observations will be very close to 1.0. This is an expected



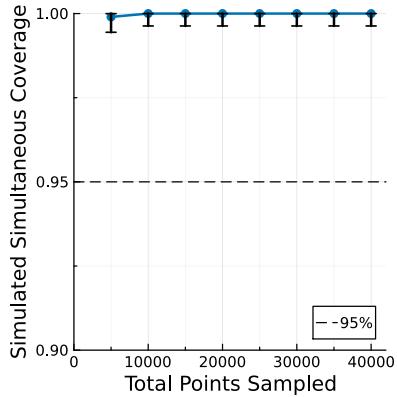
(a) Logistic simultaneous.



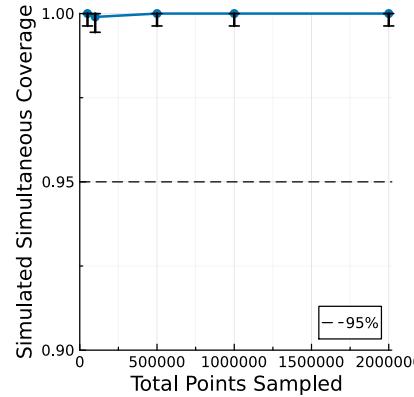
(b) Lotka-Volterra simultaneous.



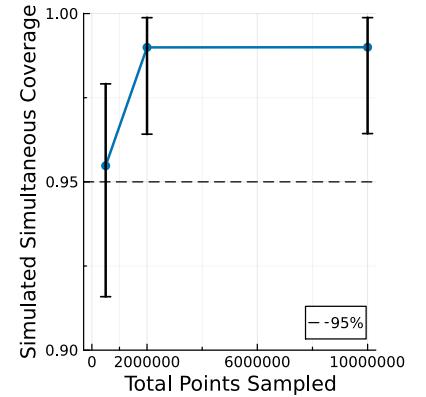
(c) Two-species logistic simultaneous.



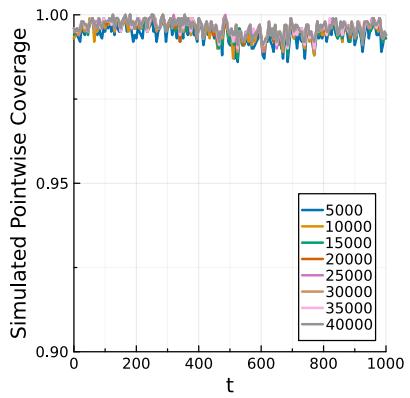
(d) Logistic simultaneous alternative.



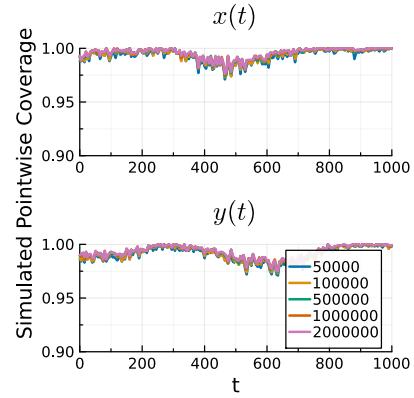
(e) Lotka-Volterra simultaneous alternative.



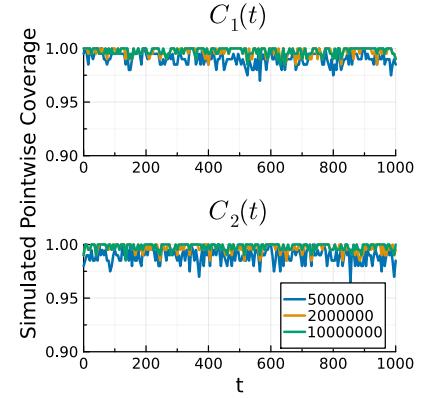
(f) Two-species logistic simultaneous alternative.



(g) Logistic pointwise.



(h) Lotka-Volterra pointwise.



(i) Two-species logistic pointwise.

Figure 30: Observation coverage of full reference tolerance sets across an increasing number of points sampled. The logistic and Lotka-Volterra models [1] used 1000 coverage simulations, while the two-species logistic model [2] used 200. Simultaneous (100% of observations captured), Simultaneous alternate (at least 95% of observations captured) and pointwise (whether an observation is captured) coverage is examined. 95% confidence intervals for the simulated coverage are provided as error bars. result as if the (95%, 95%) reference tolerance set contains the 95% reference set, it will also contain at least 95% of observations on average. Moreover, if the (95%, 95%) reference tolerance set does not contain the 95% reference set, it may still contain at least 95% of observations. This explains

the excellent performance of the full reference tolerance sets at meeting the alternative observation coverage goal.

As seen in Figure 30c, the two-species logistic model has markedly better simultaneous coverage of observations than the other models, reaching up to 0.75 [0.68, 0.81] at ten million samples. Nonetheless, this is still significantly lower than 0.95. The pointwise coverage for both observed quantities is consistently close to 1.0 for ten million samples (Figure 30i), explaining the increase in simultaneous observation coverage relative to the previous models. As for the other models, the alternative observation coverage goal is above the desired 95% coverage at 0.99 [0.96, 1.0] (Figure 30f).

Therefore, the primary aim of the $(1 - \delta, 1 - \alpha)$ reference tolerance set approach in observation space can be summarised as capturing the $1 - \delta$ reference set with probability $1 - \alpha$. Moreover, as shown here, a secondary aim of this approach can be summarised as capturing $1 - \delta$ of observations with at least a probability of $1 - \alpha$ (but likely much more than that). Furthermore, the desired coverage of this alternative observation goal can be reached on these models when using full reference tolerance sets.

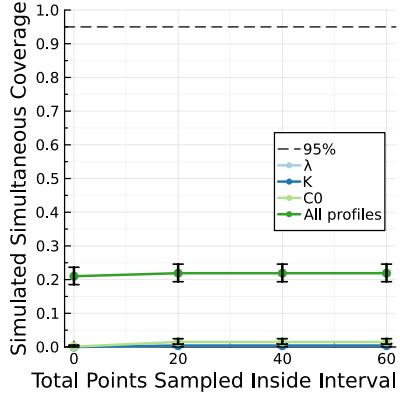
5.2 Univariate Profiles

We now consider the performance of profile-wise prediction sets from univariate profiles on the same models as in Section 5.1. The PWA workflow is now being used. We consider the impact on coverage from sampling points along each parameter profile within the parameter confidence interval, the coverage of individual profiles and the coverage of the union of all profiles. We expect the pointwise coverage of profile-wise trajectory confidence sets and profile-wise reference tolerance sets from univariate parameter profiles to be highly variable across time. The variability will depend on how influential each parameter is for quantifying uncertainty in each location in the prediction space. For example, the long-term population density trajectory (steady state) in the two logistic models considered is highly dependent on the univariate profile of the carrying capacity density, K [1, 2]. As a result, the profile-wise prediction sets from this parameter's profile will initially have low pointwise coverage that increases as the model approaches a steady state.

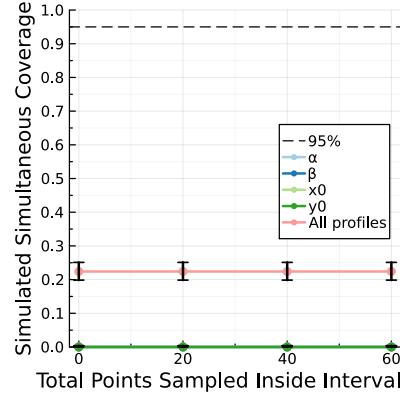
5.2.1 Profile-Wise Trajectory Confidence Set Coverage

Figure 31 shows the coverage of profile-wise trajectory confidence sets from univariate profiles on each model. In only the logistic model, sampling 20 linearly spaced points along the profile within the parameter confidence interval helps improve the simultaneous coverage of sets from individual profiles by a little (≈ 0.0 to ≈ 0.008). Regardless, the simultaneous coverage of each profile's

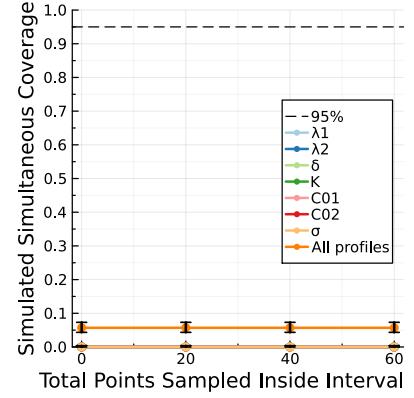
trajectory confidence set is effectively 0.0 on each model. This is not surprising given the dips to less than 0.2 pointwise coverage seen in the pointwise plots for the sets from each profile.



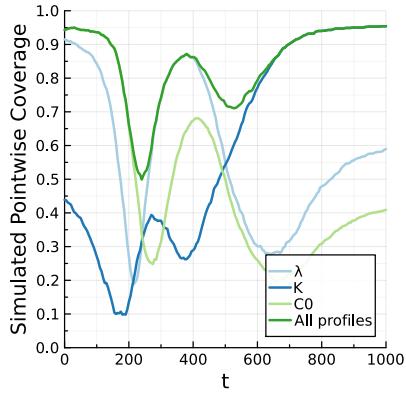
(a) Logistic simultaneous.



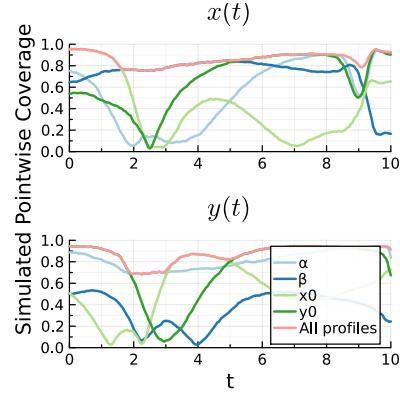
(b) Lotka-Volterra simultaneous.



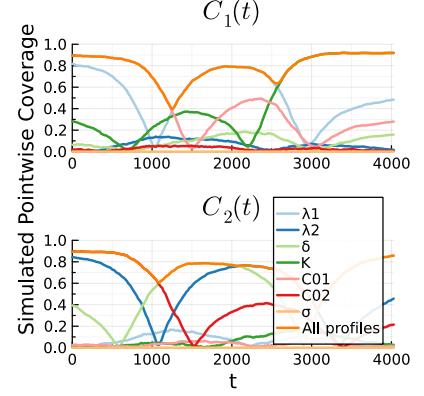
(c) Two-species logistic simultaneous.



(d) Logistic pointwise.



(e) Lotka-Volterra pointwise.



(f) Two-species logistic pointwise.

Figure 31: Profile-wise trajectory confidence set coverage from individual and the union of univariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Pointwise plots are from profiles with 60 additional points sampled inside each confidence interval. 95% confidence intervals for the simulated coverage are provided as error bars.

The pointwise coverage of the profile-wise trajectory confidence set from each univariate profile demonstrates the importance of various parameters in capturing uncertainty in the model solution at different times. For example, the profile-wise trajectory confidence set for the carrying capacity density, K , in the logistic model has pointwise coverage > 0.9 for $t \in [700, 1000]$ (Figure 31d). For $t < 700$, the pointwise coverage drops closer to 0.4.

We also note that the profile-wise trajectory confidence set for the data distribution standard deviation parameter, σ , in the two-species logistic model, has pointwise coverage of 0.0 (Figure 31f). This is because the values of σ across the profile have no impact on the model trajectory, with all nuisance parameters, ω , set to their corresponding MLE value, $\hat{\theta}^M$. Instead, it only impacts the profile log-likelihood values observed.

Therefore, to better approximate the full trajectory confidence set, we take the union across all profile-wise trajectory confidence sets formed from univariate profiles [1]. When doing this the simultaneous coverage improves to 0.22 [0.19, 0.25], 0.22 [0.20, 0.25] and 0.06 [0.04, 0.07] on the respective models in Figures 31a 31b and 31f. This is still relatively low and drops off for more than four parameters. The pointwise coverage for this union is improved in Figure 31, with coverage of at least the maximum of any individual profile at each timepoint. However, there are still several dips below 0.7 in each model. Additionally, the pointwise coverage in each model is not close to being consistently above a value of 0.975, which Section 5.1.1 indicated is required to get coverage of the model trajectory of around 0.95.

The low simultaneous and pointwise coverage of the model trajectory for the profile-wise trajectory confidence set from the union of univariate profiles motivates using bivariate profiles to better capture uncertainty [1]. Additionally, the coverage of this profile-wise trajectory confidence set appears to get worse for models with more parameters. There is an excellent chance this is directly related to the asymptotic threshold used to create the univariate profile's confidence set. Namely, the degrees of freedom parameter, ν , used to calibrate the asymptotic threshold, is set to 1 for univariate profiles. It is instead set to $|\theta|$ for full parameter confidence sets. This means that the extremities of the profile paths in full parameter space do not touch the extremities of the full parameter confidence set for $|\theta| > 1$, and these get further away as $|\theta|$ increases. For example, at a 95% confidence level and $\nu = 1$, $\ell_c = -\Delta_{1,0.95}/2 \approx -1.92$, while for $\nu \in \{2, 3, 4, 7\}$, $\ell_c \approx \{-3.00, -3.91, -4.74, -7.03\}$. Resultantly, an element of any improved coverage observed for profile-wise trajectory confidence sets from bivariate profiles may be because the asymptotic threshold used is more than 50% larger ($| -3. | > | -1.92 \times 1.5 |$).

5.2.2 Profile-Wise Reference Tolerance Set Coverage

As for the full parameter vector confidence set (Section 5.1.2), the profile-wise reference tolerance set coverage from univariate profiles in the logistic and Lotka-Volterra model is identical to the profile-wise trajectory confidence set coverage. Similarly, the profile-wise reference tolerance set coverage slightly differs on the two-species logistic model, as in Section 5.1.2, because the data distribution standard deviation is estimated.

Coverage in the two-species logistic model can be seen in Figure 32. The most interesting differences here relative to the simultaneous coverage of the model trajectory in Figure 31c are: the coverage of the profile-wise reference tolerance set from the union of all profiles has improved from 0.06 [0.04, 0.07] to 0.21 [0.19, 0.24] and the coverage of the profile-wise reference tolerance set from σ has improved

from 0.0 to 0.09 [0.07, 0.11]. This indicates that σ is the most important parameter for capturing uncertainty in the $1 - \delta$ population reference set. This makes sense as this parameter directly defines the width of each reference interval. Nonetheless, the simultaneous coverage of these profile-wise reference tolerance sets is still far from the desired value of 0.95. As seen for profile-wise trajectory confidence set coverage in this model, sampling additional points within the interval does not improve the simultaneous coverage.

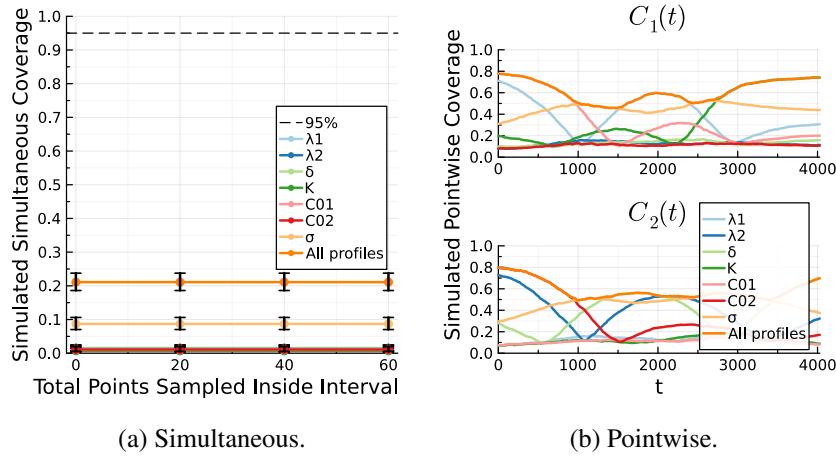


Figure 32: Profile-wise reference tolerance set coverage from individual and the union of univariate profiles in the two-species logistic model with logit-normal data distribution [2] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

The pointwise coverage in Figure 32b helps to explain the differences between the simultaneous coverage of profile-wise trajectory confidence sets and profile-wise reference tolerance sets on this model. Most significantly, the pointwise coverage from σ is no longer 0.0; it is steadily above 0.4. Additionally, the pointwise coverage from all other profiles is more consistent, with the differences between the maximums and minimums ‘squashed’. The minimums are now consistently above 0.1 instead of hovering around 0.0. This is likely to be because of the dependence of the reference tolerance set on σ ; when the profile-wise trajectory confidence set coverage is poor, to compensate, the value of σ may tend to be larger than its true value, allowing the profile-wise reference tolerance set coverage to improve and vice versa.

While the pointwise coverage of the $1 - \delta$ population reference set from the union of univariate profiles is generally more consistent than the pointwise coverage of the model trajectory, it has dropped on average. Simultaneous coverage has increased despite this. This increase means that if a single $1 - \delta$ population reference interval is contained in the profile-wise reference tolerance set, the chance of all $1 - \delta$ population reference intervals being included has increased, relative to the chance of all trajectories being included in a profile-wise trajectory confidence set if a trajectory at a single

timepoint is included. This is, again, likely explained by the dependence of the profile-wise reference tolerance set on σ .

The upper endpoint of the confidence interval for σ provides the simultaneous coverage of this profile's reference tolerance set (0.9 [0.7, 0.11]). As discussed in Section 5.2.1, all values of ω along the profile are set to $\hat{\theta}^M$. Resultantly, the extremities of the approximate $(1 - \delta, 1 - \alpha)$ reference tolerance set from σ are defined by evaluating the $1 - \delta$ reference set of the MLE solution with a higher than MLE value of σ (the upper endpoint). This presents an alternative, computationally inexpensive approach to estimating the $1 - \delta$ population reference set, similar to using $\hat{\theta}$ alone as in Murphy, Maclaren, and Simpson [9]. However, the exact value is not fully realised here, as the simultaneous coverage of this single profile is still much lower than 0.95. Section 6.2.2 provides a commentary on how the additional value of this approach could be realised.

We find evidence that the degrees of freedom parameter, ν , is constraining the obtained coverage when considering the simultaneous coverage of the $1 - \delta$ population reference set of around 21% on the two-species logistic model. As stated in Section 5.2.1, the univariate threshold at a 95% confidence level is $\ell_c = -\Delta_{1, 0.95}/2 \approx -1.92$. Similarly, the value of the 21% confidence level threshold with $\nu = |\theta| = 7$ is $\ell_c = -\Delta_{7, 0.21}/2 \approx -1.96$. This is so close that it is unlikely to be a coincidence; the threshold used appears to constrain the coverage we could get in this example. However, this exact consistency between values of ℓ_c at the obtained coverage is not observed for profile-wise prediction sets from the union of univariate profiles on the other two models (i.e. 22% confidence level with $\nu = 3$ has $\ell_c = -\Delta_{3, 0.22}/2 \approx -0.54$ and 23% with $\nu = 4$ has $\ell_c = -\Delta_{4, 0.23}/2 \approx -0.91$). We will continue investigating this for bivariate profiles in Section 5.3.

5.2.3 Observation Coverage

The coverage of observations, minus simultaneous coverage, for profile-wise reference tolerance sets from univariate profiles, can be seen in Figure 33. We do not include figures for the simultaneous coverage in this section, as it was already well below 0.95 for full parameter confidence sets (Section 5.1.2) when the full reference tolerance set coverage was around 0.95. Resultantly, the far worse coverage of these profile-wise reference tolerance sets intuitively manifests as worse simultaneous coverage of observations.

Interestingly, the coverage of the alternative simultaneous goal from the profile-wise reference tolerance set of each univariate profile is quite good, with at least 0.75 coverage for the three-parameter logistic and four-parameter Lotka-Volterra models. Additionally, the coverage of this goal from the union of univariate profiles in the first two models is above 0.95 (0.97 [0.96, 0.98] and 0.995 [0.99,

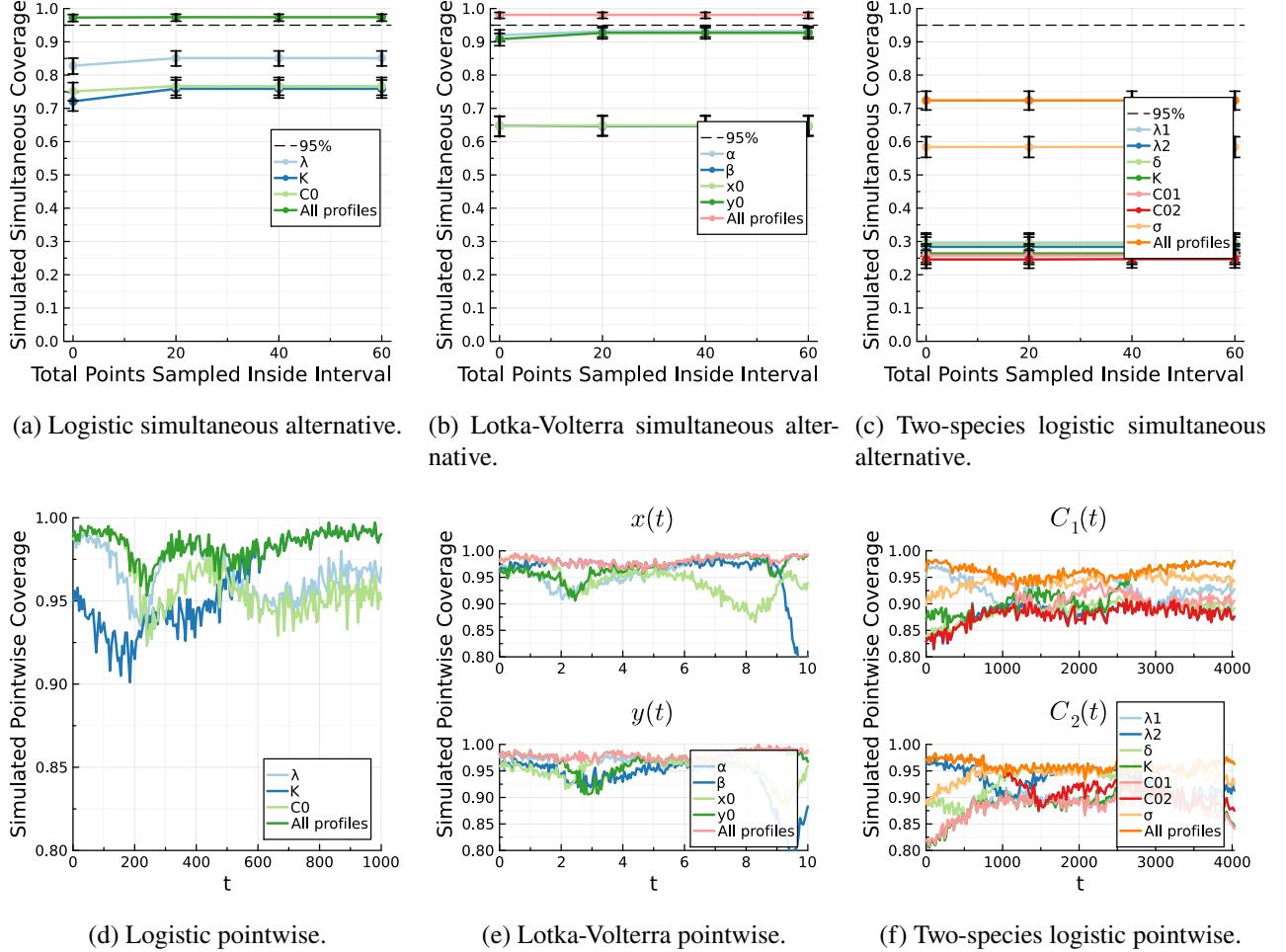


Figure 33: Observation coverage of profile-wise reference tolerance sets from individual and the union of univariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Simultaneous alternate (at least 95% of observations captured) and pointwise (whether an observation is captured) coverage is examined. Pointwise plots are from profiles with 60 additional points sampled inside each confidence interval. 95% confidence intervals for the simulated coverage are provided as error bars.

1.0]). However, this drops to a low 0.72 [0.70, 0.75] for the union of all univariate profiles for the two-species logistic model. This indicates that using profile-wise reference tolerance sets from the union of univariate profiles could be useful for this particular metric, but the coverage is still affected by $|\theta|$. This again motivates the consideration of bivariate profiles in Section 5.3.

5.3 Bivariate Profiles

We now consider the performance of profile-wise prediction sets from bivariate profiles on the same models as in Section 5.1. We consider the impact on coverage of sampling additional points inside the boundary of each profile, the coverage of individual profiles and the coverage of the union of all profiles. Because we capture the interaction between two parameters instead of only one, we expect the pointwise coverage of profile-wise prediction sets from bivariate profiles to be higher and less

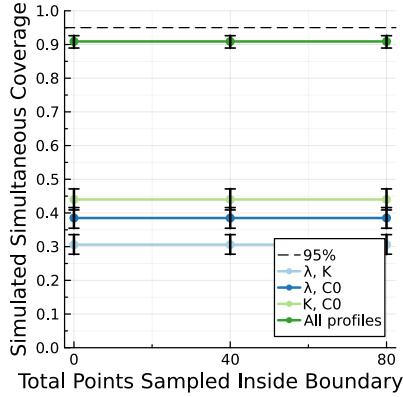
variable over time than those from univariate profiles. Resultantly, profile-wise prediction sets from the union of bivariate profiles are expected to have better simultaneous properties than the union of univariate profiles as well [1]. However, we might expect that their coverage is again non-ideal relative to those from the full parameter confidence set because the degrees of freedom parameter used for computing bivariate profiles, $\nu = 2$, is again lower than the $\nu \in \{3, 4, 7\}$, used for the full parameter confidence set in the respective considered models.

We note that the number of combinations of bivariate profiles is a combinatorial problem, $\binom{|\theta|}{2}$, which explodes in size for higher numbers of parameters. It is highly likely that only a specific subset or some smaller random subset of bivariate profiles is required to produce profile-wise prediction sets with similar coverage to profile-wise prediction sets from the union of all bivariate profiles. Similarly, these subsets may still reach a very high level of coverage. However, we will not consider this in more detail in this section; this will be discussed in Section 6.3.

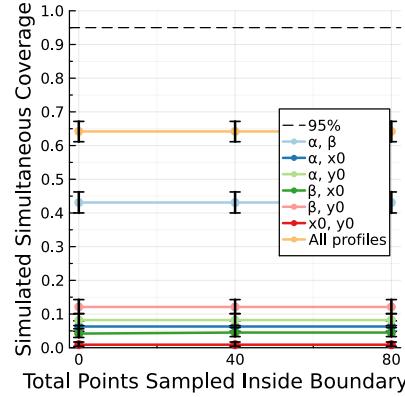
5.3.1 Profile-Wise Trajectory Confidence Set Coverage

The coverage of profile-wise trajectory confidence sets from bivariate profiles can be seen in Figure 34 on each model. Sampling additional internal points within each profile boundary using the Latin Hypercube approach detailed in Section 4.2 either does not improve or barely improves the simultaneous coverage of the model trajectory when considering individual profiles; $x(0), y(0)$ in the Lotka-Volterra model shows a slight improvement from 0.047 [0.035, 0.062] to 0.056 [0.042, 0.072] when sampling 40 points inside the boundary (Figure 34b). It does not impact the simultaneous coverage of the profile-wise trajectory confidence set from the union of profiles. Therefore, this demonstrates that it was acceptable in Simpson and Maclare [1] to assume that the boundary points of the bivariate profiles approximately map to the extremities of the profile-wise trajectory confidence set in these models. However, this still may not hold in general.

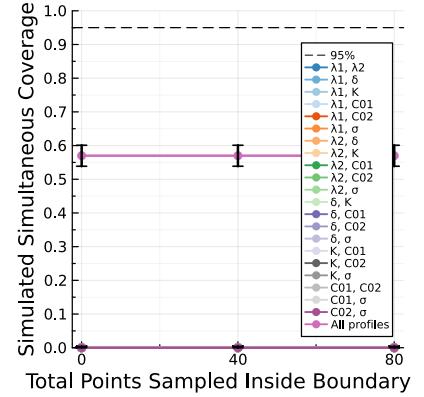
As expected, we note a significant improvement in the simultaneous coverage of the profile-wise trajectory confidence set from the union of bivariate profiles on each model in comparison to the same set from the union of univariate profiles in Section 5.2.1. The performance of profile-wise trajectory confidence sets from individual bivariate profiles in the logistic and Lotka-Volterra models is also improved, with values as high as 0.45 instead of always ≈ 0.0 . The simultaneous coverage in the respective models improves from 0.22 [0.19, 0.25], 0.23 [0.21, 0.26] and 0.06 [0.04, 0.07] to 0.91 [0.89, 0.93], 0.64 [0.61, 0.67] and 0.57 [0.54, 0.60]. However, this is still below the desired coverage of 0.95 obtained by the full trajectory confidence sets (Section 5.1.1). Additionally, the coverage gets worse as the number of model parameters increases.



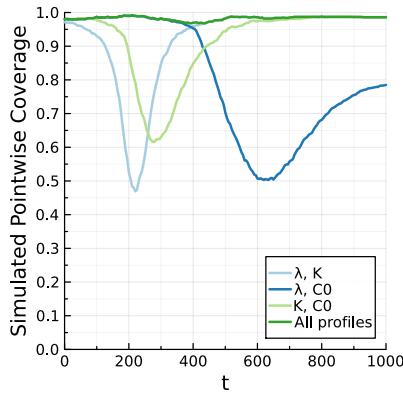
(a) Logistic simultaneous.



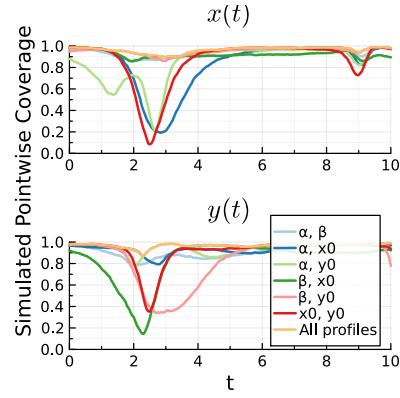
(b) Lotka-Volterra simultaneous.



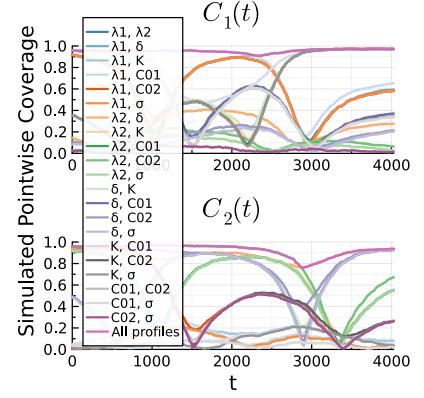
(c) Two-species logistic simultaneous.



(d) Logistic pointwise.



(e) Lotka-Volterra pointwise.



(f) Two-species logistic pointwise.

Figure 34: Profile-wise trajectory confidence set coverage from individual and the union of bivariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Pointwise plots are from profiles with zero additional points sampled inside each boundary. 95% confidence intervals for the simulated coverage are provided as error bars.

This again indicates that the asymptotic threshold used for these bivariate profiles constrains the simultaneous coverage of the model trajectory that could be reached if ν was closer to $|\theta|$. The bivariate asymptotic threshold at a 95% confidence level is $\ell_c = -\Delta_{2,0.95}/2 \approx -3.00$. Similarly, the value of the 91% confidence threshold with $\nu = |\theta| = 3$ in the logistic model is $\ell_c = -\Delta_{3,0.91}/2 \approx -3.25$. These two thresholds are very close. Meanwhile, in the Lotka-Volterra and two-species logistic model the 72% and 57% thresholds with $\nu = |\theta|$ are $\ell_c \approx \{2.54, 3.49\}$. While these thresholds are not as close to the bivariate threshold, they are similar enough to indicate that the bivariate threshold used contributes to the lower-than-desired coverage.

The pointwise coverage of profile-wise trajectory confidence sets from bivariate profiles does tend to be higher than those from univariate profiles (Figure 31), with fewer large drops seen in Figure 34. For example, the profile-wise trajectory confidence set from λ, K in the logistic model (Figure 34d) only has a single dip down to 0.47 from a consistent value of 0.97. In contrast, the profile-wise

trajectory confidence sets from these parameters individually had multiple larger dips down to as much as 0.1 and never reached more than 0.95. This will be mainly due to the bivariate profiles capturing the interaction between two parameters, which allows the profiles to capture more uncertainty in model trajectory space relative to univariate profiles. It can also be partially attributed to the higher asymptotic threshold used.

We observe that specific profiles are more important for estimating one of the components of the model solution than the other. For example, the pointwise coverage of the profile-wise trajectory confidence set from the $\beta, y(0)$ profile of the $x(t)$ component in the Lotka-Volterra model is typically around 0.95, while it has a large dip to around 0.35 for the $y(t)$ component around $t = 3.0$ (Figure 34e) This will be useful to consider when looking at the subsets of profiles needed to reach high levels of coverage in Section 6.3.

5.3.1.1 Impact of Observations of Important Features

The pointwise coverage in the Lotka-Volterra model also indicates another issue; pointwise coverage drops on at least one model component for $t \in [1.5, 4]$ occur for the profile-wise trajectory confidence sets from every bivariate profile. Considering the plot of the true model trajectory in Appendix A.2, we can see this is associated with the following feature. The prey species, $x(t)$, hits a population peak at $t \approx 2$, followed by a subsequent decline as the predator species, $y(t)$, hits a population peak at $t \approx 3$, also followed by a subsequent decline as the population of prey falls. A similar (small) dip on $x(t)$ can be observed for profile-wise trajectory confidence sets from several of these profiles for $t \approx 9$, where this feature occurs again. This indicates that the coverage properties of profile-wise prediction sets, even at a bivariate level, can struggle if there is insufficient information on important features like this one. With the observations available, predicting this feature will likely depend on more than just two interest parameters. The issue emphasises the approximate nature of the profile-wise prediction sets from bivariate profiles; the full trajectory confidence set in Section 5.1.1 did reach simultaneous coverage of 0.95 on this model.

It also emphasises the importance of sampling enough observations of features that strongly depend on more than two parameters, allowing nuisance parameter values along the bivariate profiles to approach their relevant true values. For example, the location of the peak in each model component at this early time strongly depends on the relative values of α and β . This is why this is the only profile with a profile-wise trajectory confidence set with relatively high simultaneous coverage (0.43 vs less than 0.15); it is the best at predicting the trajectory of both model components simultaneously. Good coverage of both components is an expected result given the pointwise coverage of each model

trajectory component for profile-wise trajectory confidence sets from the profiles of α , which is good for $x(t)$, and β , which is good for $y(t)$ (Figure 31e). The reasonably good pointwise coverage of the model trajectory from this profile indicates that the nuisance parameter values for the initial value of each model component are often reasonably consistent with their true values along the α, β bivariate profile. Meanwhile, in the other bivariate profiles, which have worse pointwise coverage of the model trajectory at this location, the values of these nuisance parameters are likely to be much less consistent with their true values.

Therefore, to resolve this issue for profile-wise trajectory confidence sets from bivariate profiles, we could either sample more points around this problematic time or another set of timepoints where the feature occurs again. The feature occurs again at late time ($t \in [8, 10]$), so sampling this region should have a similar effect. This will be considered in Section 6.3.1.

5.3.2 Profile-Wise Reference Tolerance Set Coverage

As in previous sections, we only discuss the profile-wise reference tolerance set coverage from bivariate profiles in the two-species logistic model. This can be seen in Figure 35. The profile-wise reference tolerance set from the union of bivariate profiles shows improved coverage of 0.57 [0.54, 0.60] over the set from the union of univariate profiles (0.21 [0.19, 0.24], Section 5.2.2). This is identical (to two d.p.) to the coverage of the model trajectory from the union of bivariate profiles observed on this model in Section 5.3.1. Nonetheless, this is still much lower than the desired coverage of 0.95. Again, we see no improvement in the simultaneous coverage of any profile when sampling 40 points inside the boundary.

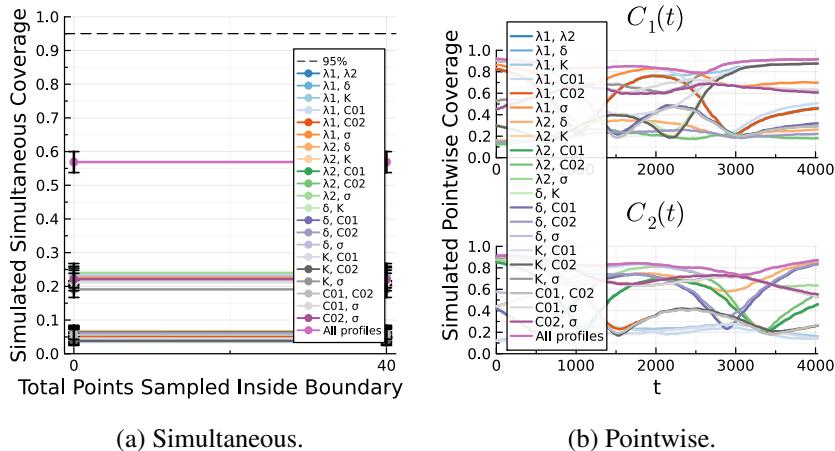


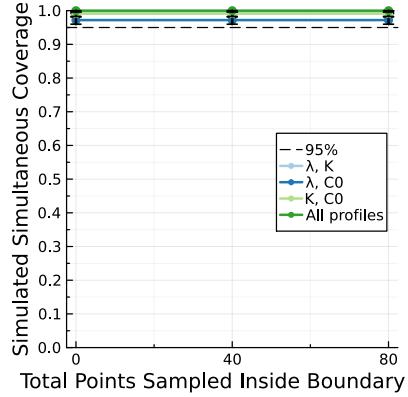
Figure 35: Profile-wise reference tolerance set coverage from individual and the union of bivariate profiles in the two-species logistic model with logit-normal data distribution [2] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

As observed for the profile-wise reference tolerance sets from univariate profiles on this model

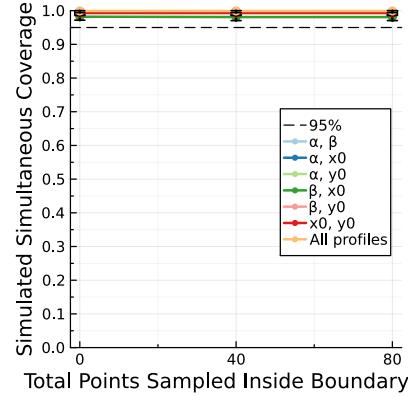
(Section 5.2.2), sets from bivariate profiles that contain σ have markedly better simultaneous coverage of the $1 - \delta$ population reference set (≈ 0.23 instead of ≈ 0.06). This indicates that profile-wise reference tolerance sets from this subset of profiles may be the most important for reaching high reference tolerance set coverage levels when θ^o is estimated. For bivariate profiles, as observed for univariate profiles, the pointwise coverage of profile-wise reference tolerance sets relative to profile-wise trajectory confidence sets is more consistent and ‘squashed’, with minimums raised from 0.0 to 0.15 and maximums decreased from 0.97 to 0.9. As stated in Section 5.2.2, this is likely because of the dependence of the reference tolerance set on σ .

5.3.3 Observation Coverage

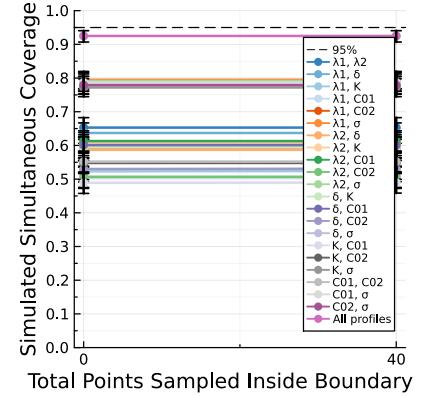
The coverage of observations, minus simultaneous coverage, for profile-wise reference tolerance sets from bivariate profiles, can be seen in Figure 36. The coverage of the alternative simultaneous goal for profile-wise reference tolerance sets from each profile in the first two models is well above 0.95, with the union of all profiles at approximately 1.0. The coverage of profile-wise reference tolerance sets from individual profiles in the two-species logistic model that does not contain σ tends to be around 0.6, while those that do are around 0.78. This model has coverage from the union of bivariate profiles at 0.93 [0.91, 0.94]. This is a much-improved result from the low 0.72 [0.70, 0.75] seen from propagating forward the union of univariate profiles in the two-species logistic model (Figure 33c). The improved performance relative to univariate profiles is down to the pointwise coverage being consistently close to 1.0 in each model. This indicates that the union of these profiles could be useful for this particular metric. However, the union of these profiles does appear to have reduced coverage of this goal as $|\theta|$ increases. Again, this may indicate that the asymptotic threshold constrains this coverage as $|\theta|$ increases.



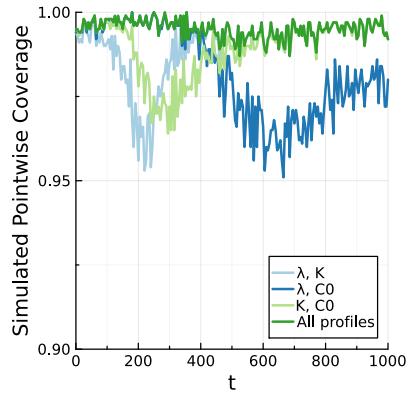
(a) Logistic simultaneous alternative.



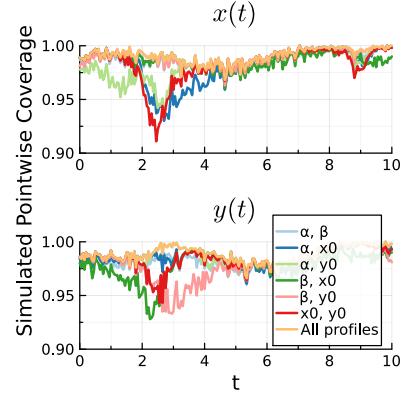
(b) Lotka-Volterra simultaneous alternative.



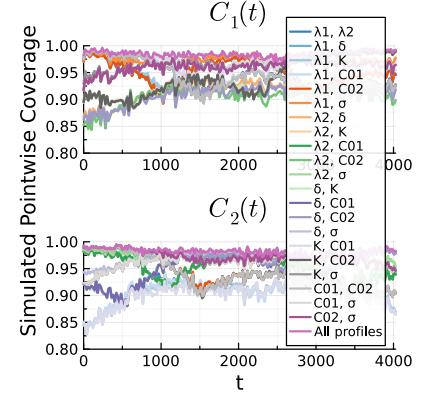
(c) Two-species logistic simultaneous alternative.



(d) Logistic pointwise.



(e) Lotka-Volterra pointwise.



(f) Two-species logistic pointwise.

Figure 36: Observation coverage of profile-wise reference tolerance sets from individual and the union of bivariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Simultaneous alternate (at least 95% of observations captured) and pointwise (whether an observation is captured) coverage is examined. 95% confidence intervals for the simulated coverage are provided as error bars.

6 Profile Paths For Profile-Wise Analysis: Coverage of Predictive Quantities on Simple Models

In this section, we consider the impact on the coverage of using the simultaneous asymptotic threshold, $\ell_c = -\Delta_{|\theta|, 1-\alpha}/2$, to produce profile-wise sets for predictive quantities propagated forward from profiles. This has not yet been explored within the PWA workflow; the results we obtained in Section 5 indicated that setting the threshold based on the dimensionality of the interest parameter of the profile was constraining the observed coverage performance. Resultantly, we expect coverage to improve at the very least and, at best, reach similar levels to prediction sets formed from full parameter confidence sets, at significantly lower computational cost. We will refer to this updated approach as *profile paths* for profile-wise analysis; in these simultaneous profiles, we try to evaluate good ‘paths’ of points along the extremities of the full parameter confidence set. In doing so, we will consider the performance of random subsets of profiles and hand-selected subsets. We will introduce a discussion on how to identify these profile subsets systematically. Finally, we will compare our results for all profile-wise prediction sets to those from full parameter confidence sets in Section 5.1.

If the boundary of a univariate or bivariate profile defines the extremities of the set for a predictive quantity, as generally found for the models discussed in Sections 5.2 and 5.3, then it is reasonable to expect that the coverage of these sets is tied to the asymptotic threshold used to define the profile’s boundary contour. Furthermore, this becomes more relevant as higher dimensional models are considered because more dimensions increase the distance between the profile’s boundary and the full parameter confidence set. Therefore, using the simultaneous asymptotic threshold should allow the coverage of profile-wise prediction sets formed from the profile paths procedure to approach the coverage of prediction sets from the full parameter confidence set more quickly. The coverage is not expected to be identical because the threshold is not solely responsible for the low coverage. For example, we would expect low coverage occasionally for profile-wise trajectory confidence sets formed from simultaneous univariate profiles, given our existing discussion of the dependence of the model trajectory on sets of multiple parameters across time (Section 5.3.1).

In this section, we only consider this approach on identifiable models with low numbers of parameters (up to seven). This limits the applicability of any conclusions of the value of the profile-path approach for models with non-identifiability or large numbers of parameters. In particular, we envisage issues where confidence interval widths blow up for very high numbers of parameters. Additionally, non-identifiable parameters potentially should not contribute to the degrees of freedom of the threshold. For instance, structurally non-identifiable parameters which provide no new information

on predictive quantities. We recommend repeating this analysis on non-identifiable models and higher dimensional models, such as those in Simpson and Maclaren [16], Borisov and Metelkin [22], and Boehm et al. [116], to assess the effectiveness of the approach under different conditions. The exact settings used may need to be tweaked on these models if their non-identifiability produces issues when using the ellipse approximation as a starting solution. For example, it may be best to start `IterativeBoundaryMethod` from `RadialRandomMethod` rather than `RadialMLEMethod`.

6.1 Profile Path Example

To demonstrate the value of the profile path approach, we return to the example introduced in Section 2.2.8.1; a single parameter scalar model, from which we have obtained $I = 100$ observations corrupted by i.i.d. Gaussian noise. The main idea of the approach is for lower dimensional profiles to trace a good path through higher dimensional space, where the boundary of the profile touches the boundary of the higher dimensional space.

The contours of the log-likelihood function corresponding to the univariate asymptotic threshold, $\ell_c = -\Delta_{1,0.95}/2$, and the simultaneous asymptotic threshold, $\ell_c = -\Delta_{2,0.95}/2$ can be seen in Figure 37a. This figure also plots the interest parameter and corresponding nuisance parameter of univariate profiles at each threshold. We see that univariate profiles at the univariate threshold (light, dashed green and red) only extend until the inner contour, while those at the simultaneous threshold (dark, solid green and red) extend until the outer contour. This allows the simultaneous univariate profiles to touch the boundary of the full parameter confidence set.

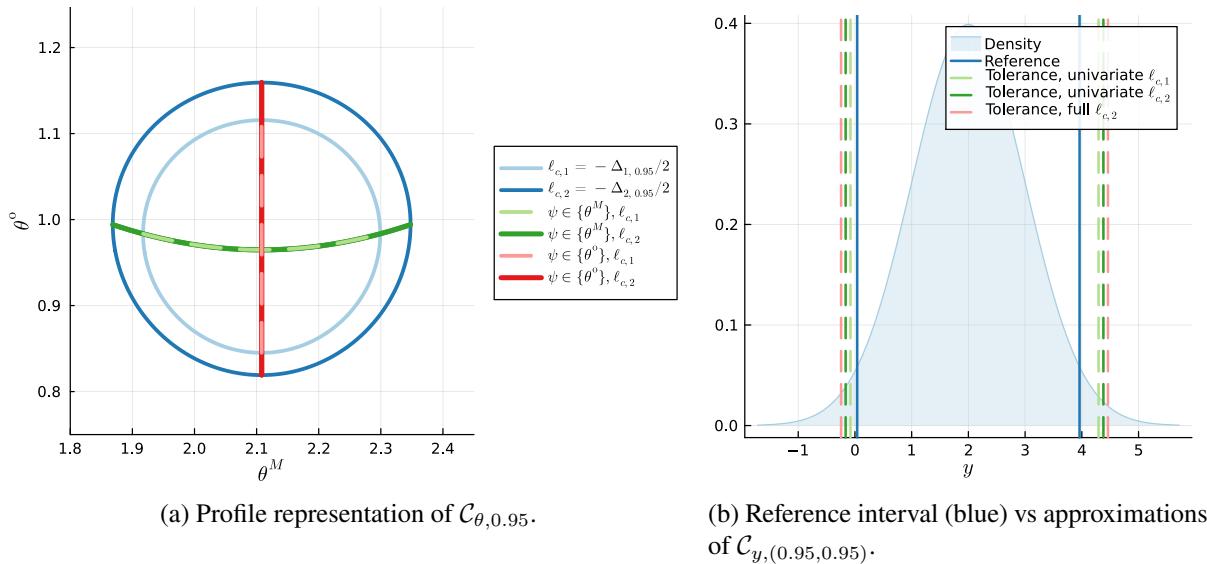


Figure 37: Profile-wise reference tolerance interval approximations, $\mathcal{C}_{y,(0.95,0.95)}$, from univariate and bivariate (full) profiles at univariate and simultaneous asymptotic thresholds, given 100 observations of y .

This threshold difference results in a corresponding difference in the coverage properties and width of profile-wise sets for predictive quantities. In Figure 37b, all the approximate (95%, 95%) reference tolerance intervals contain the 95% population reference interval. However, the profile-wise reference tolerance interval from the union of simultaneous univariate profiles is a closer approximation of the full reference tolerance interval. The approximation is not exact due to the lack of knowledge of other locations along the boundary (other parameter interactions).

When simulating the coverage properties of the profile-wise reference tolerance intervals from each of these profiles, the improved approximation offered by the simultaneous univariate profiles is significant. The results of 2000 coverage simulations are given in Table 7. The full reference tolerance interval (bivariate profile) had coverage of more than 95% (0.98 [0.98 0.99]). Similarly, the profile-wise reference tolerance interval from the union of simultaneous univariate profiles approached this level at 0.96 [0.95, 0.96]. Most significantly, the coverage observed for the profile-wise reference tolerance interval from the union of univariate profiles at the univariate threshold failed to reach 95% (0.89 [0.88, 0.90]). Therefore, evaluating lower dimensional profiles at the simultaneous asymptotic threshold can effectively improve the statistical performance of the profile-wise approximation.

	Univariate union	Univariate union	Bivariate
Asymptotic threshold	$\ell_c = -\Delta_{1, 0.95}/2$	$\ell_c = -\Delta_{2, 0.95}/2$	$\ell_c = -\Delta_{2, 0.95}/2$
Coverage	0.890 [0.875, 0.903]	0.956 [0.946, 0.964]	0.984 [0.977, 0.989]

Table 7: Profile-wise reference tolerance set coverage of the union of univariate and bivariate profiles evaluated at the univariate and simultaneous asymptotic thresholds in the Gaussian noise model from 2000 coverage simulations. 95% confidence intervals for the simulated coverage are bracketed.

6.2 Simultaneous Univariate Profiles

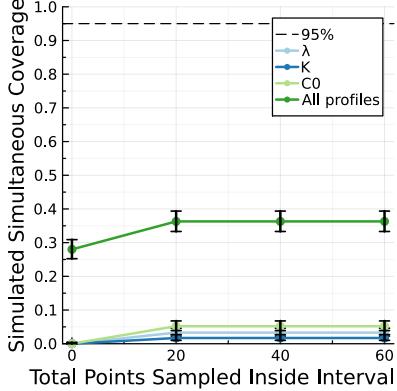
We now consider the performance of profile paths for profile-wise prediction sets from univariate profiles on the same models as in Section 5.1. These univariate profiles are formed using the simultaneous asymptotic threshold, $\ell_c = -\Delta_{|\theta|, 0.95}/2$: they are simultaneous univariate profiles. As in Section 5.2, we consider the impact on coverage from sampling points along each parameter profile within the parameter confidence interval, the coverage of individual profiles and the coverage of the union of all profiles. We expect to see the coverage performance of profile-wise prediction sets from simultaneous univariate profiles improve relative to those made using profiles at the univariate threshold. This is because the higher threshold produces wider univariate profiles than those at the univariate threshold, incorporating more uncertainty in parameter space. Nonetheless, we still expect to see some variability in coverage over time (although less).

6.2.1 Profile-Wise Trajectory Confidence Set Coverage

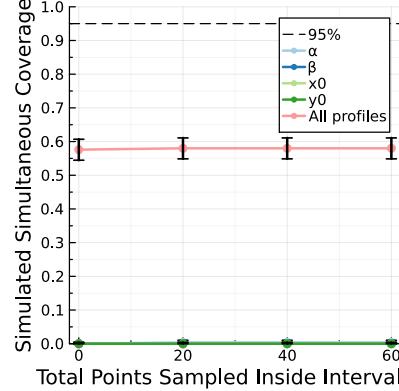
The coverage of profile-wise trajectory confidence sets from simultaneous univariate profiles can be seen in Figure 38 on each model. The unions from the simultaneous univariate profiles are a good, albeit still imperfect, improvement on the unions of univariate profiles. The simultaneous coverage of the model trajectory improves from 0.22 [0.19, 0.25], 0.22 [0.20, 0.25] and 0.06 [0.04, 0.07] in Section 5.2.1 to 0.36 [0.33, 0.39], 0.58 [0.54, 0.61] and 0.43 [0.40, 0.46] on each model, respectively. Again, in only the logistic model, sampling 20 linearly spaced points along the profile helps improve the simultaneous coverage of each individual simultaneous profile and their union. This helps improve the simultaneous coverage of these individual profiles from around 0.008 to around 0.03. However, this is still very low. Similarly, the simultaneous coverage of the individual simultaneous profiles in the other models is effectively 0.0. This is, once again, not surprising as the dips in the pointwise coverage of the model trajectory have not improved significantly. However, pointwise coverage has increased in each model on average, contributing to the improved simultaneous coverage of the union of simultaneous univariate profiles. Additionally, the profile for σ still has pointwise coverage of 0.0.

The points within the interval result may indicate that sampling points within the interval is only important for making profile-wise prediction sets when the model parameter dimension, $|\theta| < \approx 4$. However, sampling points inside each simultaneous confidence interval is relatively cheap, so it is generally recommended. This does not necessarily indicate that non-boundary points of the full parameter confidence set are required for accurate predictions. Instead, it may indicate that more boundary points are needed; points along the profile which improve coverage may be close to boundary points in full parameter space that are at least as good or better than these points. This is supported by our findings with bivariate profiles at the simultaneous threshold; their boundary points appear to be generally sufficient for driving the extremities of sets for predictive quantities on these models (Section 6.3.1).

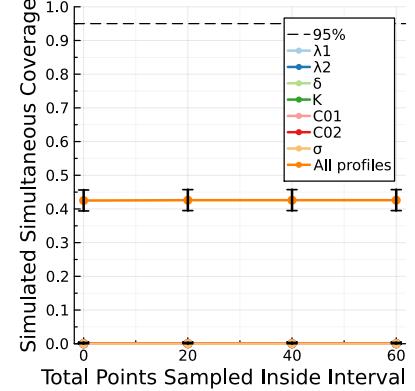
Although the simultaneous coverage of the model trajectory from the union of all profiles in each model disappointingly fails to reach 0.95, inspection of the pointwise coverage tells a slightly different story. In the logistic model in Figure 38d, there are two wide dips in pointwise coverage down to 0.7 and 0.8 for $t \approx 220$ and $t \approx 550$, respectively, which constitutes about 40% of the time considered. However, pointwise coverage is closer to 0.95 or higher outside of these times. Similarly, for the Lotka-Volterra model in Figure 38e, there are several wide dips to around 0.90. These dips also only occur for around 30% of the time considered and are located at the feature discussed in Section 5.3.1; outside of these dips, pointwise coverage is closer to 0.98. This is also observed for the two-species logistic model in



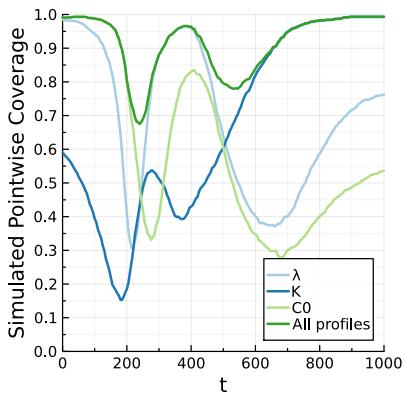
(a) Logistic simultaneous.



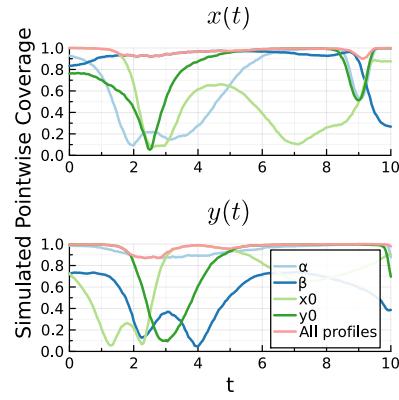
(b) Lotka-Volterra simultaneous.



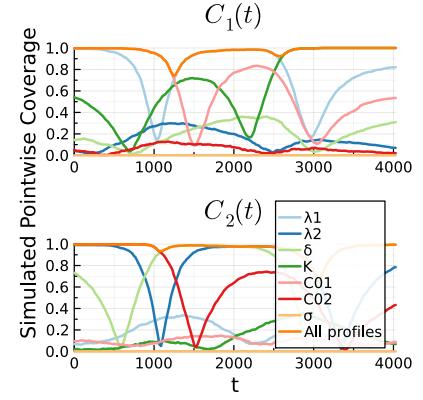
(c) Two-species logistic simultaneous.



(d) Logistic pointwise.



(e) Lotka-Volterra pointwise.



(f) Two-species logistic pointwise.

Figure 38: Profile-wise trajectory confidence set coverage from individual and the union of simultaneous univariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Pointwise plots are from profiles with 60 additional points sampled inside each confidence interval. 95% confidence intervals for the simulated coverage are provided as error bars.

Figure 38f. Therefore, the coverage of the model trajectory from the union of simultaneous univariate profiles is fairly good, particularly for models with higher numbers of parameters.

The pointwise coverage plots contain useful sensitivity information which can be used to understand each model better. For instance, the times when the coverage of the union of simultaneous univariate profiles is high tend to correspond to times when the uncertainty in the model trajectory is effectively dependent on the uncertainty in a single dominant parameter. The model trajectory at these times effectively has a one-dimensional approximation as expected [1]. These times can be seen as the pointwise coverage for all profiles touching the pointwise coverage for each single profile at the relevant timepoints. For example, the profile-wise trajectory confidence set coverage in the logistic model (Figure 38d) is entirely dependent on the initial condition, $C(0)$, at early time ($t \in [0, 180]$), the growth rate, λ , at mid-time ($t \in [260, 400]$), and the carrying capacity, K , at late-time ($t \in [650, 1000]$). This

is a significant reason why the profile-wise approach was put forward. Namely, different parameters will drive the model trajectory at certain times [1]. Resultantly, it is reasonable to expect that the union of simultaneous univariate profiles can capture a large amount of the uncertainty in predictive quantities. This is seen to be the case in each of the models considered here.

Similarly, the dips in the pointwise coverage tend to correspond to times when the coverage of the profile-wise trajectory confidence set from the profile driving the model trajectory decreases. These dips can be explained as times when the model trajectory depends on multiple parameters. In some cases, this is when the trajectory transitions from being dominated by a particular parameter to another parameter. In the logistic model, and to a lesser extent, the other models, these times can be seen as the pointwise coverage of the set from the union of simultaneous profiles being higher than the pointwise coverage of the set from any single profile. For example, the profile-wise trajectory confidence set coverage in the logistic model from the union of simultaneous univariate profiles (Figure 38d) sees dips between the periods of dominance of each of the parameters. In this case, it appears that the model trajectory is dependent on the dominant parameter in the time before and after each dip, i.e. $(\lambda, C(0))$ for early to mid-time and (λ, K) for mid time to end time. This supports the conclusion that considering the interaction between at least two parameters is important for predicting the model trajectory. This will be further discussed when considering bivariate profiles at the simultaneous threshold in Section 6.3.1.

6.2.2 Profile-Wise Reference Tolerance Set Coverage

In Figure 39, the profile-wise reference tolerance set coverage from simultaneous univariate profiles is significantly improved for the two-species logistic model relative to that of the profile-wise trajectory confidence set. The profile-wise reference tolerance set from the union of simultaneous univariate profiles has a simultaneous coverage of 0.83 [0.80, 0.85], which is much higher than the 0.21 [0.19, 0.24] coverage from the same union at the univariate threshold in Section 5.2.2. Additionally, the set from this simultaneous univariate profile union has simultaneous coverage of effectively the same level as that obtained by two million samples of the full parameter confidence set (0.85 [0.80, 0.90], Section 5.1.1). This is a great result, as the full parameter confidence set used better-informed parameter bounds yet still required around 35 times more evaluations (two million vs around 56,900 in Appendix K.1, Table 11).

The profile-wise reference tolerance set from the profile for σ has simultaneous coverage of 0.61 [0.57, 0.64] from its upper endpoint alone, which explains the significant improvement in coverage. Its coverage is much higher than the simultaneous coverage of around 0.15 observed for sets from

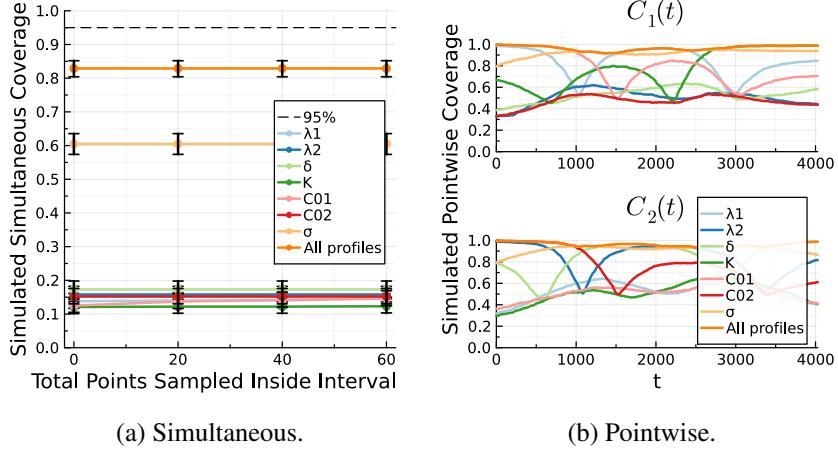


Figure 39: Profile-wise reference tolerance set coverage from individual and the union of simultaneous univariate profiles in the two-species logistic model with logit-normal data distribution [2] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars. every other profile. As discussed in Section 5.2.2, this demonstrates that σ is the most important parameter for capturing uncertainty in the $1 - \delta$ population reference set.

The pointwise coverage is again ‘squashed’ compared to the pointwise coverage of the profile-wise trajectory confidence set (Figure 38f). However, unlike the profile-wise reference tolerance set from the union of univariate profiles, which had a maximum of 0.8 with dips down to 0.4, the set from the union of simultaneous univariate profiles has pointwise coverage consistently above 0.92 for each model component. The improvement in the minimums for this set relative to the trajectory confidence set is once again likely to be because of the dependence of the reference tolerance set on σ (Section 5.2.2).

6.2.3 Observation Coverage

The coverage of observations, excluding simultaneous coverage, for profile-wise reference tolerance sets from simultaneous univariate profiles can be seen in Figure 40. The coverage of the alternative simultaneous observation goal is effectively at 1.0 from the union of simultaneous univariate profiles in each model. This is an improvement on the coverage in the two-species logistic model from the union of univariate profiles, which did not reach more than 0.72 [0.70, 0.85] (Section 5.2.3). Additionally, the cost of evaluating profiles at the simultaneous and univariate thresholds is not significantly different (Appendix K.1). Therefore, if this goal is of interest, we can cheaply make predictions for observations from the union of simultaneous univariate profiles.

The alternative simultaneous coverage of observations goal is largely unaffected by sampling points inside each simultaneous confidence interval. However, that does not mean it is not improving coverage of observations; the improvements in simultaneous coverage of the model trajectory seen for

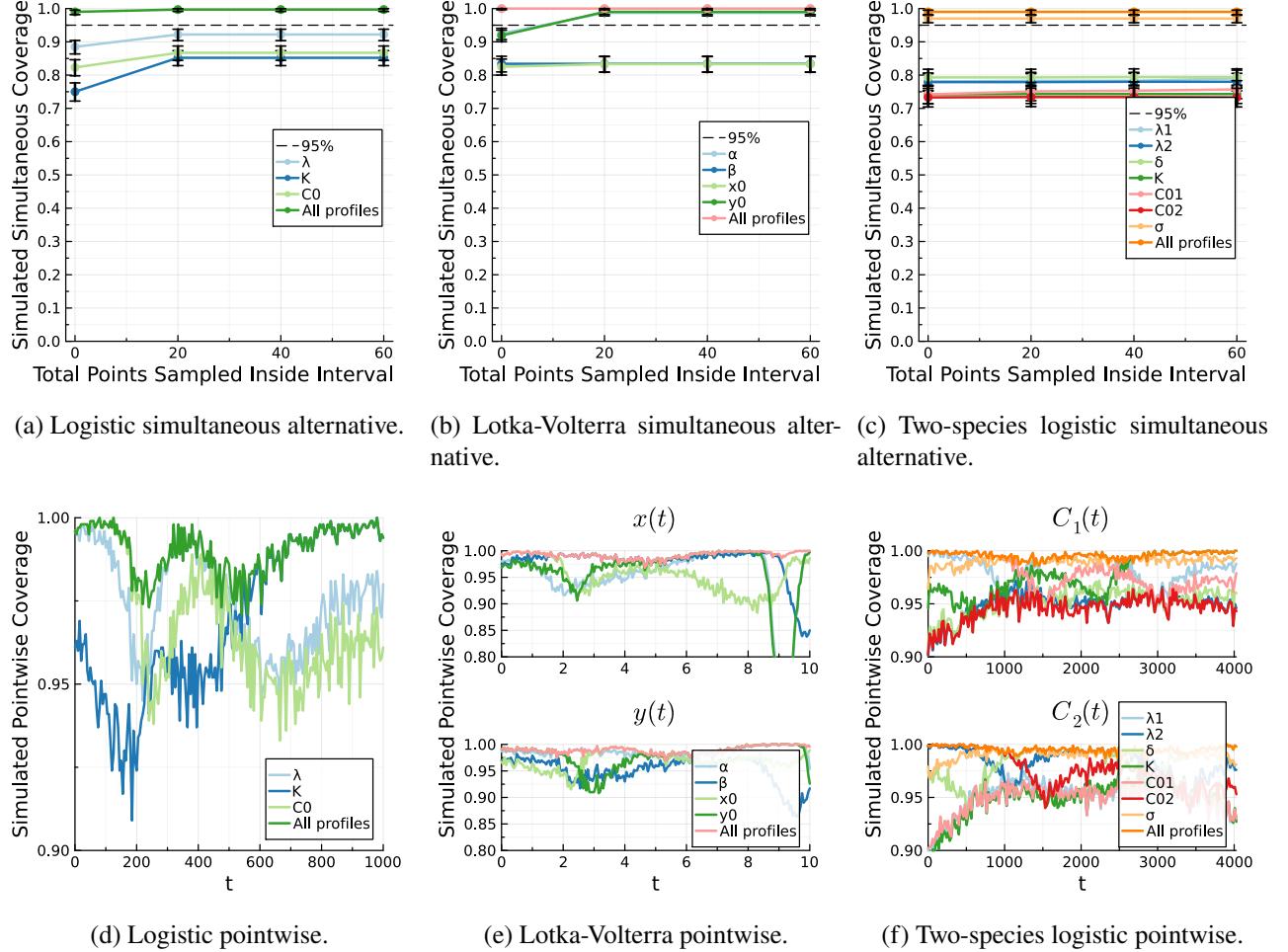


Figure 40: Observation coverage of profile-wise reference tolerance sets from individual and the union of simultaneous univariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Simultaneous alternate (at least 95% of observations captured) coverage is examined. Pointwise plots are from profiles with 60 additional points sampled inside each confidence interval. 95% confidence intervals for the simulated coverage are provided as error bars.

the logistic model in Section 6.2.1 will also correspond to higher pointwise coverage of observations. Resultantly, we still generally recommend sampling points within the interval for observation coverage.

6.3 Simultaneous Bivariate Profiles

We now consider the performance of profile paths for profile-wise predictions from bivariate profiles on the same models as in Section 5.1. These bivariate profiles are formed using the simultaneous asymptotic threshold, $\ell_c = -\Delta_{|\theta|, 0.95}/2$: they are simultaneous bivariate profiles. As in Section 5.3, we consider the impact on coverage of sampling additional points inside the boundary of each profile, the coverage of individual profiles and the coverage of the union of all profiles.

We also consider the performance of profile-wise prediction sets from random subsets of simultaneous bivariate profiles and investigate the existence of a small specific subset that obtains a similar level of

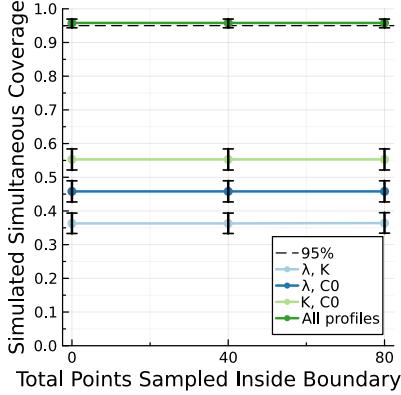
coverage to the sets from the union of all profiles. These subsets are critical when considering higher dimensional systems. In this case, evaluating all simultaneous bivariate profiles may be computationally infeasible or unnecessary for quantifying uncertainty in predictive quantities. For example, a 100-parameter system has $\binom{100}{2} = 4950$ combinations, where perhaps only 100 combinations are required for uncertainty quantification. We will introduce a discussion on methods that could be used to identify this subset systematically.

6.3.1 Profile-Wise Trajectory Confidence Set Coverage

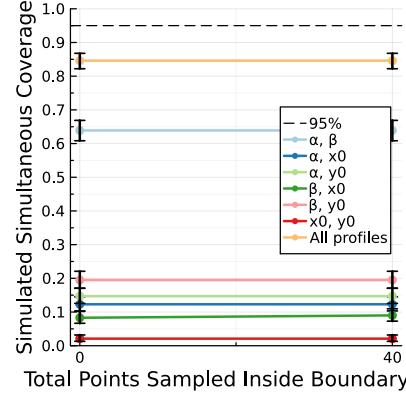
The coverage of profile-wise trajectory confidence sets from bivariate profiles at the simultaneous threshold can be seen in Figure 41 on each model. As seen for sets from bivariate profiles at the bivariate threshold (Section 5.3.1), no significant improvement in the simultaneous coverage is seen from sampling points inside the boundary of bivariate profiles. This supports the conclusion in Section 6.2.1 that sampling points along the univariate profile in the logistic model is only required because too few points on the boundary of the full parameter confidence set have been found. Additionally, those points we are sampling along the boundary of this set using simultaneous bivariate profiles are generally sufficient for quantifying the uncertainty in the model trajectory. Namely, in each model, the simultaneous coverage of profile-wise trajectory confidence sets from the union of simultaneous bivariate profiles is either close to or is 0.95.

The coverage of profile-wise trajectory confidence sets from bivariate profiles at the simultaneous threshold can be seen in Figure 41 on each model. As seen for sets from bivariate profiles at the bivariate threshold (Section 5.3.1), no significant improvement in the simultaneous coverage is seen from sampling points inside the boundary of bivariate profiles. This supports the conclusion in Section 6.2.1 that sampling points along the univariate profile in the logistic model is only required because too few points on the boundary of the full parameter confidence set have been found. Additionally, those points we are sampling along the boundary of this set using simultaneous bivariate profiles are generally sufficient for quantifying the uncertainty in the model trajectory. Namely, in each model, the simultaneous coverage of profile-wise trajectory confidence sets from the union of simultaneous bivariate profiles is either close to or is 0.95.

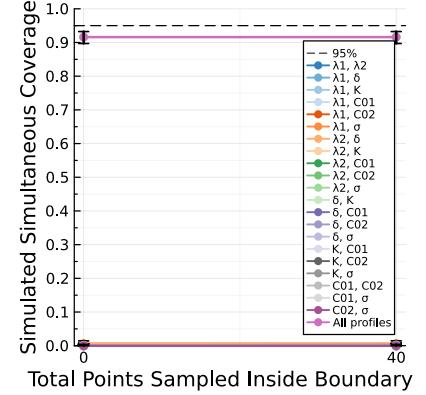
The simultaneous coverage of the model trajectory is greatly improved in Figure 41 for the profile-wise trajectory confidence set from the union of simultaneous bivariate profiles in comparison to the set from the union of bivariate profiles in Section 5.3.1. On each model respectively, it improves from 0.91 [0.89, 0.93], 0.72 [0.69, 0.74] and 0.57 [0.54, 0.60] to 0.95 [0.94, 0.96], 0.85 [0.82, 0.87] and 0.92, [0.90, 0.93]. The desired simultaneous coverage of the model trajectory is therefore observed for



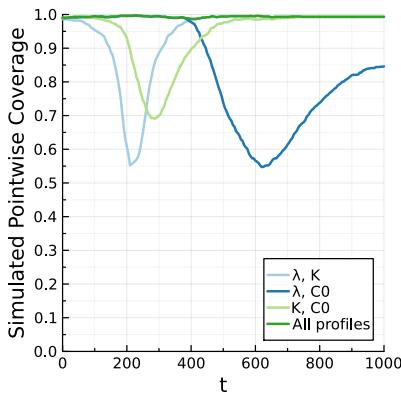
(a) Logistic simultaneous.



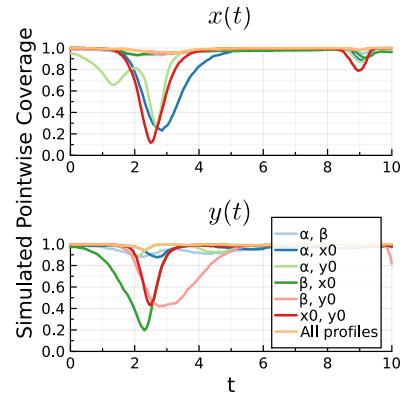
(b) Lotka-Volterra simultaneous.



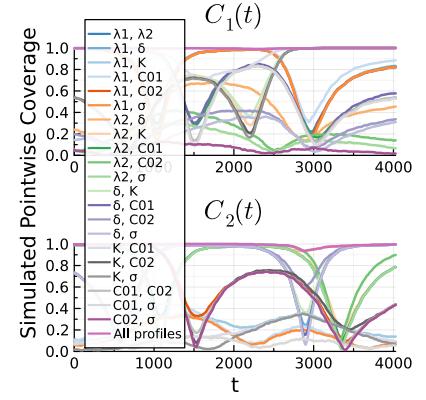
(c) Two-species logistic simultaneous.



(d) Logistic pointwise.



(e) Lotka-Volterra pointwise.



(f) Two-species logistic pointwise.

Figure 41: Profile-wise trajectory confidence set coverage from individual and the union of simultaneous bivariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Pointwise plots are from profiles with zero additional points sampled inside each boundary. 95% confidence intervals for the simulated coverage are provided as error bars.

the logistic model when using the simultaneous threshold. The reduction in simultaneous coverage for increasing $|\theta|$ is also not present.

These simultaneous coverage results are a significant improvement on the results from the union of simultaneous univariate profiles in Section 6.2.1 where the highest coverage observed was 0.58 [0.54, 0.61]. Additionally, these prediction sets are still an approximation of the full trajectory confidence set, so we only expect their coverage to asymptotically approach that of the full set as $|\psi|$ approaches $|\theta|$.

In the Lotka-Volterra model, the lower than 0.95 simultaneous coverage from the union of simultaneous bivariate profiles is explained by the shallow wide dip in the pointwise graph of the $x(t)$ component for $t \in [2, 4]$; outside of this time, the pointwise coverage is excellent. This corresponds to the feature discussed in Section 5.3.1, where there are peaks in the $x(t)$ and then $y(t)$ solutions. At this time, the model trajectory is likely to depend on the interaction of more than two parameters. The full

parameter confidence set did have simultaneous coverage of 0.95 (Section 5.1.1), so increasing $|\psi|$ is likely required to improve coverage in these locations. Alternatively, sampling additional observations of this feature would reduce the uncertainty in the true model trajectory at these times, accomplishing the same result without moving away from bivariate profiles.

For example, suppose we include six additional timepoints for $t \in [7, 10]$ (Appendix A.2), in which the feature is observed again. In that case, the simultaneous coverage of the union improves to 0.88 [0.86, 0.90] in Figure 42, with typically excellent pointwise coverage. This also reduces the dependence on the α, β profile, which sees a reduction in simultaneous coverage from 0.64 [0.61, 0.66] to 0.43 [0.40, 0.46]. The other profiles see a corresponding improvement in simultaneous coverage, with the $\alpha, y(0)$ and $\beta, x(0)$ profiles improving to a similar level as the α, β profile. The most significant indication that sampling more of this feature improves overall coverage is seen in Figure 42b: the pointwise coverage of the set from the union of simultaneous bivariate profiles no longer has a dip for the $x(t)$ component around $t \in [2, 4]$. Additionally, this is no longer *exclusively* the location of the dips in coverage of sets from individual profiles (Figure 41e); these dips are now spread across time. This demonstrates the importance of sampling features that depend on many parameters when making predictions from bivariate profiles. Doing so will enable the nuisance parameters of simultaneous bivariate profiles to be more consistent with the values required for good simultaneous coverage.

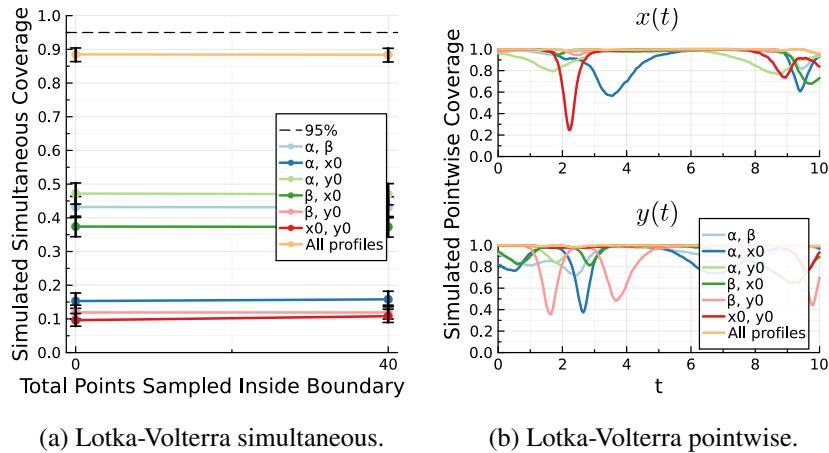


Figure 42: Trajectory confidence set coverage of individual bivariate parameter profiles and the union of all profiles using the simultaneous threshold in the Lotka-Volterra model [1] with observations timepoints $t \in [0, 10]$ from 1000 coverage simulations. Pointwise plots are from profiles with zero additional points sampled inside each boundary. 95% confidence intervals for the simulated coverage are provided as error bars.

In the two-species logistic model, the simultaneous coverage of the profile-wise trajectory confidence set from the simultaneous bivariate profile union (0.92, [0.90, 0.93]) overlaps with the coverage obtained from ten million evaluations of the full parameter confidence set, which used better-informed

parameter bounds (0.90 [0.84, 0.93] Section 5.1.1). These intervals overlapping is a great result, as the bivariate profiles represent a significant decrease in computational effort (1000 coverage simulations of the 21 bivariate profiles with the current settings took around 4.5 hours while 200 simulations of the ten million samples took around 4 hours; the workload was distributed across ten CPU threads). Moreover, if the parameter bounds used to evaluate the simultaneous bivariate profiles were also used for the full parameter confidence set, the number of samples would need to be significantly increased to reach the same coverage. The lower-than-desired coverage here can be explained by the slight dip in pointwise coverage of $C_2(t)$ for $t \approx 2800$; again, the pointwise coverage is excellent outside of this time.

These examples demonstrate that using the simultaneous threshold when forming bivariate profiles for prediction is sensible and worthwhile, at least on these models where the ‘correct’ value for the degrees of freedom, ν , is expected to be $|\theta|$. Its use appears reasonable for identifiable models with fewer than ten parameters. For models with a higher number of parameters or non-identifiable parameters, particularly structurally non-identifiable, this may be less clear and requires further work.

6.3.1.1 Reducing Computational Cost While Maintaining Coverage

The pointwise graphs for coverage of the model trajectory (Figures 41d, 41e and 41f) indicate that not all simultaneous bivariate profiles are required to create a profile-wise trajectory confidence set with a similar level of simultaneous coverage to a set from the union of all profiles. Additionally, we need more than one profile; the simultaneous coverage of sets from individual profiles is much lower than 0.95 and falls, on average, with increasing $|\theta|$. For example, the pointwise coverage in the logistic model (Figure 41d) confirms the statement from Section 6.2.1 that only two bivariate combinations are needed. In particular, these are $(\lambda, C(0))$ for early to mid-time and (λ, K) for mid-time to end-time. The pointwise coverage of the set from all profiles is equal to the pointwise coverage of the set from the first of these combinations for $t \in [0, 400]$. Similarly, it is equal to the pointwise coverage of the set from the second of these combinations for $t \in [400, 1000]$.

These two combinations correspond to the parameters that dominate at the beginning or end of these two periods (Section 6.2.1). The interaction between the two parameters thus helps to explain the time when the trajectory transitions from being dominated by one parameter to the other. This insight could also be used to derive approximate models depending on a time of interest, particularly if the model and parameters do not have clear real-world interpretations. For example, if only an accurate understanding of uncertainty at early times is desired, then K does not matter for determining the model trajectory. Similarly, $C(0)$ does not matter as much at late time. These correspond to the

asymptotic early and late-time solutions typically derived for a logistic model [117].

Therefore, considering these pointwise plots provides a simple method of determining a good subset of profiles to reach simultaneous coverage similar to the union of all profiles. However, the cost of the coverage simulation we use here is 1000 times the cost of evaluating all simultaneous bivariate profiles. Therefore, there may only be a point in using this method to evaluate a good subset if this subset will be recomputed many times, such that the cost savings from recomputing fewer profiles outweigh the cost of the coverage simulation. Nonetheless, it demonstrates that a small, accurate subset does exist. This subset could likely be systematically identified with other, cheaper approaches.

Because we have found that sampling points within these bivariate boundaries (at least on these models) is not required when making predictions, we also may not require all of the 50 or 30 boundary points in each bivariate profile (Table 6). The boundary points found by `IterativeBoundaryMethod` have good coverage of the true boundary in these models, even for around 20 points (Section 4.3.2 and 4.4). Resultantly, fewer points will likely be sufficient for quantifying uncertainty in predictive quantities. Additionally, if only boundary points are required for quantifying uncertainty in predictive quantities, then other methods may waste computational time evaluating points that are not located near or on this boundary. Examples of these methods include random sampling, or methods like MCMC which sample a Bayesian posterior proportional to the probability of each point (the posterior is the log-likelihood function when parameter priors are uniformly distributed).

6.3.2 Profile-Wise Reference Tolerance Set Coverage

The coverage of the profile-wise reference tolerance set from simultaneous bivariate profiles in the two-species logistic model can be seen in Figure 43. Most significantly, the set from the union of simultaneous bivariate profiles has simultaneous coverage of 0.92, [0.90, 0.94], which is effectively the same as the 0.92 [0.87, 0.95] obtained by ten million samples of the full parameter confidence set in Section 5.1.2. This is also slightly higher than the simultaneous coverage observed for the profile-wise trajectory confidence set from the same profile union in Section 6.3.1, albeit with overlapping confidence intervals.

As in Section 5.3.2, we observe that profile-wise reference tolerance sets from simultaneous bivariate profiles which contain σ have markedly better simultaneous coverage of the $1 - \delta$ population reference set (≈ 0.69 instead of ≈ 0.28). This again demonstrates that $\sigma(\theta^0)$, when estimated, is the most important parameter for capturing uncertainty in the $1 - \delta$ population reference set. This is slightly higher than the 0.61 [0.57, 0.64] simultaneous coverage of the set from the simultaneous univariate profile of σ in Section 6.2.2. However, this comes at a much higher computational cost; instead

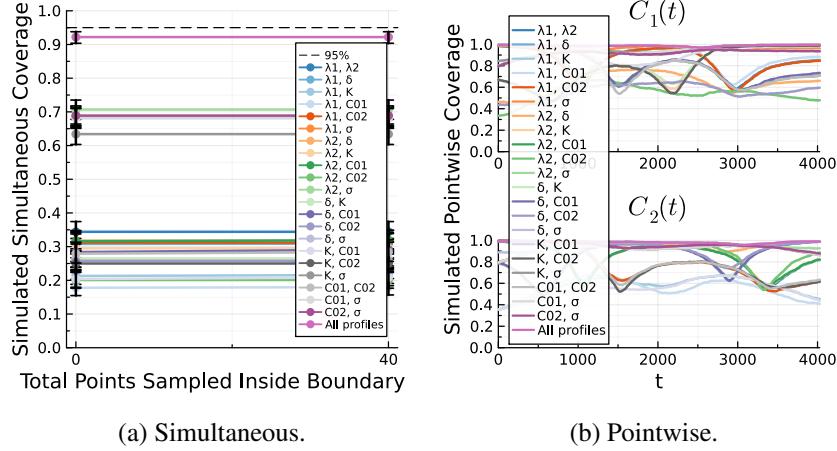


Figure 43: Profile-wise reference tolerance set coverage from individual and the union of simultaneous bivariate profiles in the two-species logistic model with logit-normal data distribution [2] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

of ≈ 2800 likelihood function calls (Appendix K.1, Table 11) a bivariate profile including σ would require ≈ 90000 calls with the current settings (Appendix K.2, Table 14).

The simultaneous coverage of the $1 - \delta$ population reference set of 0.92 [0.90, 0.94] is a much-improved result from the coverage of 0.57 [0.54, 0.60] observed for the profile-wise reference tolerance set from the union of bivariate profiles in Section 5.3.2. And, it reached as the ‘gold-standard’ full reference tolerance set it approximated. Once more, this demonstrates that evaluating the boundary of bivariate profiles at the simultaneous threshold, $\nu = |\theta|$, is sensible and worthwhile for this model. Setting $\nu = 2$ when forming bivariate profiles was constraining their simultaneous coverage of predictive quantities on this model, preventing the full value of this approximation from being realised.

We observe the same ‘squashed’ behaviour for the pointwise coverage of the $1 - \delta$ population reference set in Figure 43b in comparison to the coverage of the model trajectory in Figure 41f. The maximum pointwise coverage stays at around 1.0, while the minimum gets raised from 0.0 to around 0.35. This is a similar level of ‘squash’ to that seen for profile-wise reference tolerance sets from simultaneous univariate profiles, which had the minimum raised to around 0.3 (Section 6.2.2).

6.3.3 Observation Coverage

The coverage of observations, excluding simultaneous coverage, for profile-wise reference tolerance sets from simultaneous bivariate profiles, can be seen in Figure 44. As seen for the sets from the union of simultaneous univariate profiles in Section 6.2.3, the sets from the union of simultaneous bivariate profiles have coverage of the alternative simultaneous observation goal at effectively 1.0 in each model. We also observe a very high pointwise coverage of observations of at least 0.98. This is an improvement on the coverage of this alternative goal from bivariate profiles at the bivariate threshold

in the two-species logistic model, with little difference for the other two models (Section 5.3.3).

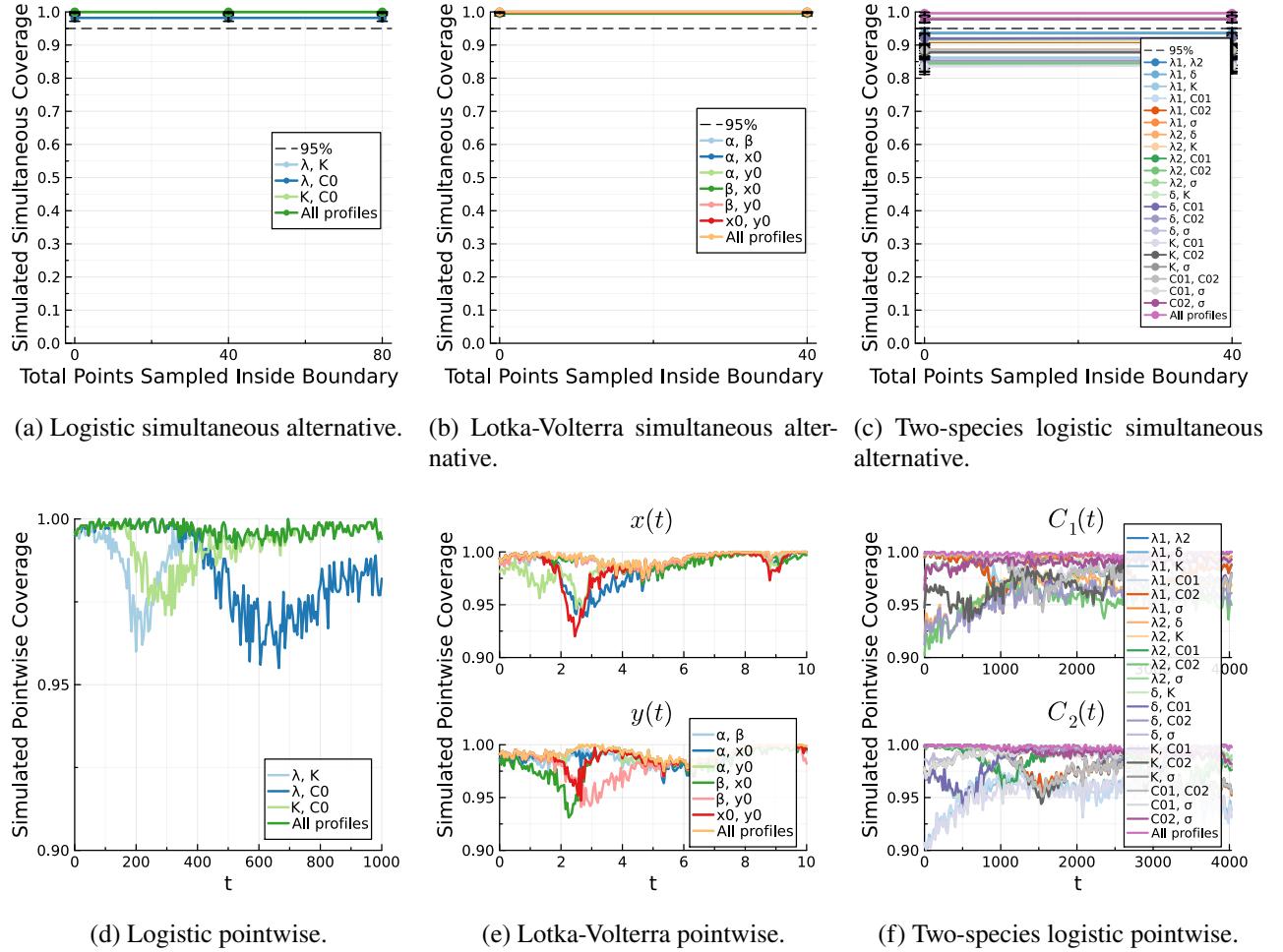


Figure 44: Observation coverage of profile-wise reference tolerance sets from individual and the union of simultaneous bivariate profiles in the logistic, Lotka-Volterra and two-species logistic models [1, 2] from 1000 coverage simulations. Simultaneous alternate (at least 95% of observations captured) coverage is examined. 95% confidence intervals for the simulated coverage are provided as error bars.

If this alternative observation coverage goal is solely of interest, then a case can be made to rely on the union of all simultaneous univariate profiles. This union has a similar computational cost to an individual simultaneous bivariate profile (see Appendix K.1 vs Appendix K.2). However, if this is not the sole goal, then using multiple simultaneous bivariate profiles is recommended.

6.3.4 Impact of Using a Subset of Profiles on Coverage

Although we can compute bivariate profiles in parallel, it is highly likely that we only require some profiles to reach the desired level of coverage (Section 6.3.1.1). Resultantly, it may be desirable from a computational standpoint to determine a ‘good’ subset of all profiles. Choosing a subset will be particularly important as $|\theta|$ increases, given that the total number of bivariate profiles is a combinatorial problem, $\binom{|\theta|}{2}$. Moreover, finding this subset may allow a significant computational

performance advantage for the PWA workflow over sets for predictive quantities formed from the sampled full parameter confidence set, at a comparable level of coverage.

One basic method for determining a subset would be to take a random subset of all the profiles without replacement. Another method that could be used is to consider the pointwise coverage of each bivariate profile in a coverage simulation and select a subset that maximises pointwise coverage over time, as introduced in Section 6.3.1.1. This is a very costly approach, but it will allow us to show that a small subset does exist with coverage close to that of the union of all profiles. This will then motivate a discussion of methods to systematically identify this subset in a low-cost fashion, which can be considered and expanded upon in future work.

We expect to see a similar phenomenon to other dimensionality reduction techniques such as principal component analysis (PCA) [118]. In particular, as we consider more bivariate profiles (components), we should initially see a significant increase in accuracy/coverage/consideration of uncertainty, with this increase decreasing in magnitude for subsequent combinations until it converges to the simultaneous coverage value obtained from the union of all profile combinations. Here, unlike PCA, we choose random profiles (components). The coverage performance shown here for these random subsets is the average-case performance.

As in the confidence interval performance comparisons with `LikelihoodProfiler` (Section 3.3), here we evaluate the number of mean likelihood function calls required for each bivariate boundary across 100 data realisations. These can be found in Appendix K.2 for each model and profile. To map these onto random subsets, we perform a simple simulation where, for each number of profiles in the subset, we choose a random subset of that size and sum the number of function calls. We perform this a large number of times (5000) and take the mean across the summed number of function calls for each subset's size.

In the first two models, where the trajectory and reference set coverage are equivalent, we will only consider the coverage of trajectory confidence sets. In contrast, we consider both metrics for the two-species logistic model.

6.3.4.1 Profile-Wise Trajectory Confidence Set Coverage on the Logistic and Lotka-Volterra Models

As expected, in Figure 45 we observe an initially large increase in the simultaneous coverage of the model trajectory that levels off as more bivariate profiles are included in the random subset for both the logistic and Lotka-Volterra models. We notice that the computational cost of increasing the number of random profiles considered is approximately linearly proportional to the number of profiles. Therefore,

finding a good subset will have a computational performance reduction approximately proportional to the fraction of all profiles used.

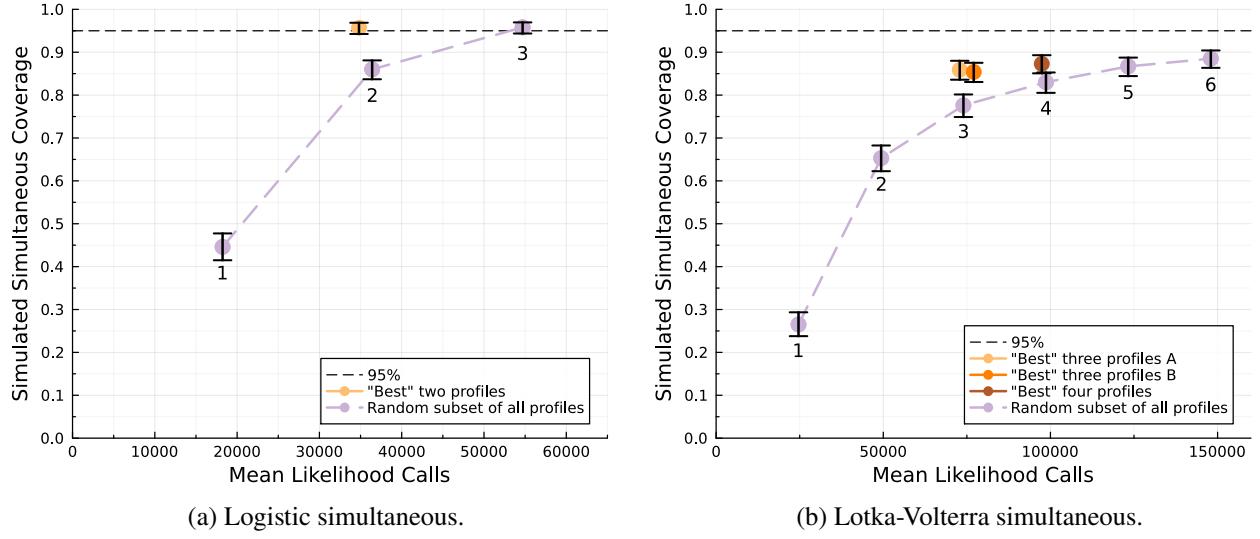


Figure 45: Mean number of likelihood function calls versus the coverage of profile-wise trajectory confidence sets from subsets of all simultaneous bivariate profiles in the logistic and Lotka-Volterra models [1] from 1000 coverage simulations. The number of random profiles in each random subset is given by the annotation below a plotted point. 95% confidence intervals for the simulated coverage are provided as error bars.

In the logistic model, the best two bivariate profiles we can select based on the pointwise coverage of the model trajectory in Figure 41d are $\psi \in \{(\lambda, K), (\lambda, C(0))\}$. The reasoning for this is covered in Section 6.3.1. The profile-wise trajectory confidence sets from the union of these two profiles enables simultaneous coverage of 0.96 [0.94, 0.97] in Figure 45a, which is equivalent to the coverage of the set from the union of all simultaneous bivariate profiles. This reduces the computational cost of this coverage (using the current settings) to approximately two-thirds of the cost of the union ($35000/55000 \approx 0.64$). Selecting bivariate profiles that use the dominant parameters for determining the model trajectory in two successive periods may be an effective method of choosing good subsets (here, they are early and middle, and middle and end).

In contrast, when considering the pointwise coverage of the model trajectory in the Lotka-Volterra model (Figure 41e), an appropriate subset is less apparent. There are now six profiles to consider over two model trajectory components, and some of these profiles perform better on one component than the other. For example, the profile for $\beta, y(0)$ exhibits good pointwise coverage of the $x(t)$ trajectory and relatively poor coverage of the $y(t)$ trajectory, with a large dip to around 0.4. It is clear that we need a combination of profiles that gets good pointwise coverage for $t \in [2, 4]$, the feature discussed in Section 5.3.1; the pointwise coverage outside of this time is much-improved for every individual

profile.

Therefore, we choose a subset of four simultaneous bivariate profiles to approach the simultaneous coverage of the model trajectory of 0.85 [0.82, 0.87] for the trajectory confidence set from the union of all profiles. For good coverage of the first model component's trajectory, we choose $\psi \in \{(\beta, x(0)), (\beta, y(0))\}$. For the second model component's trajectory, we choose $\psi \in \{(\alpha, x(0)), (\alpha, y(0))\}$. The profile $\beta, y(0)$ will also help with coverage of the second model component for $t \approx 5$. The subset chosen here makes sense from a prediction standpoint, as the feature for $t \in [2, 4]$ is highly dependent on all model parameters. Additionally, the relationship between α and β should still be captured via the optimal nuisance parameters in each profile. Resultantly, the profile-wise trajectory confidence set from this subset has simultaneous coverage of 0.84 [0.82, 0.86] in Figure 45b. This is slightly less than that seen for the set from the union of all simultaneous bivariate profiles at two-thirds of the computational cost.

We intentionally chose to omit the profile of α, β from this subset, which is the only profile with profile-wise trajectory confidence sets with good simultaneous coverage. This coverage is due to the profile having a better overall pointwise coverage on both components of the model trajectory rather than only being good at predicting a single component. However, it does not have the best pointwise coverage of each model component, so choosing other profiles which do will allow us to maximise coverage. For models with multiple trajectory components, this shows that sets from bivariate profiles with poor simultaneous coverage may be more important for prediction than sets from bivariate profiles with better simultaneous coverage.

These hand-picked subsets demonstrate that we can achieve better simultaneous coverage of the model trajectory at each subset size than randomly picking. Additionally, there is likely to be a subset that produces a profile-wise trajectory confidence set with very close to, or slightly less, simultaneous coverage than the union of all profiles for significantly less computational cost (around two-thirds in these examples). However, for this to be effective, we need a reasonable level of knowledge on which profiles to choose in the subset. If we instead chose profiles that are not complementary, meaning that each has dips in pointwise coverage at the same or very similar times, we would likely obtain results that are worse than the average random subset case.

6.3.4.2 Profile-Wise Trajectory Confidence Set and Reference Tolerance Set Coverage on the Two-Species Logistic Model

For both the coverage of the model trajectory and the $1 - \delta$ population reference set on the two-species logistic model, we observe an initially large increase in the simultaneous coverage of the

model trajectory that levels off as we include more bivariate profiles in the random subset (Figure 46). Once again, the computational cost of increasing the number of random profiles considered is approximately linearly proportional to the number of profiles. Most significantly, the coverage of profile-wise reference tolerance sets from every random subset size is much higher than that of the corresponding profile-wise trajectory confidence set. This will be due to the dependence of the reference tolerance set on σ (Section 5.2.2) which has the effect of raising the minimum pointwise coverage of the $1 - \delta$ population reference set from each profile (Figure 43b).

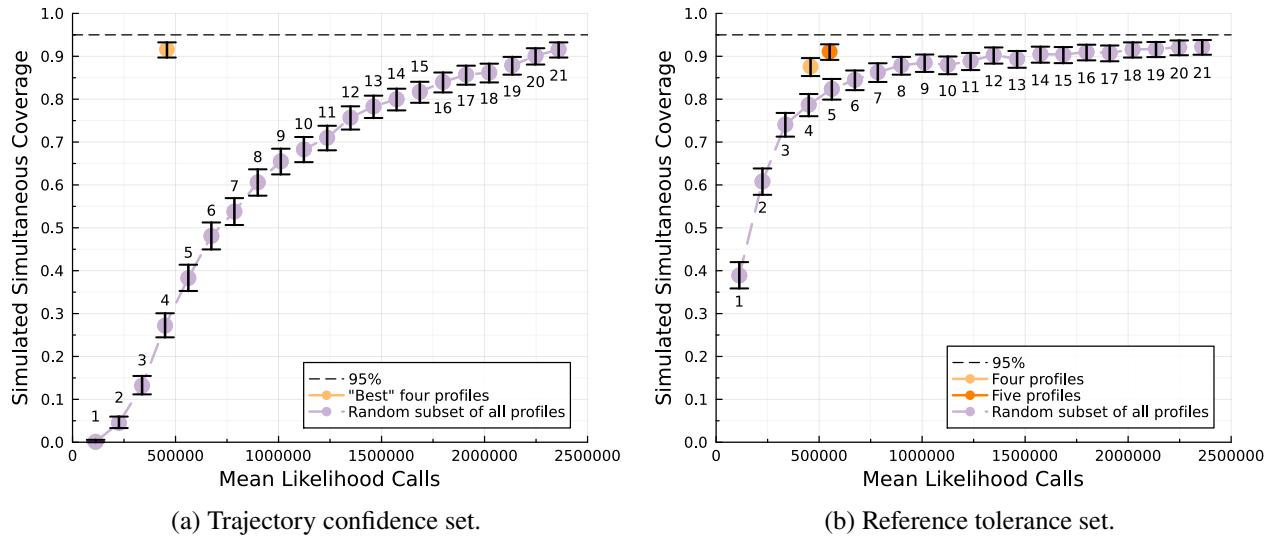


Figure 46: Mean number of likelihood function calls versus the coverage of profile-wise trajectory confidence and reference tolerance sets from subsets of all bivariate parameter profiles using the simultaneous threshold in the two-species logistic models [2] from 1000 coverage simulations. The number of random profiles in each random subset is given by the annotation below a plotted point. 95% confidence intervals for the simulated coverage are provided as error bars.

In this model, the most important bivariate profiles for producing profile-wise trajectory confidence sets with high coverage are those that directly impact the respective observed component ($C_1(t)$ or $C_2(t)$). The equations for each component are coupled by the parameter δ , which has a relatively small true value of 0.0004 in this simulation. Resultantly, the competition rate between the two coral species is relatively weak, meaning that each equation is relatively decoupled. This means that combinations of parameters that are primarily present in the equation for a given component, will have the best coverage of that component. We then form a subset from the union of these combinations, from every observed component. This contrasts the Lotka-Volterra model, in which the differential equations for the predator and prey species are highly coupled.

In particular, the four simultaneous bivariate profiles selected are $\psi \in \{(\lambda_1, K), (\lambda_1, C_1(0)), (\lambda_2, \delta), (\lambda_2, C_2(0))\}$. The first two profiles effectively allow 100% pointwise coverage of the $C_1(t)$ component,

and the last two profiles effectively allow 100% pointwise coverage of the $C_2(t)$ component. The simultaneous coverage of the profile-wise trajectory confidence sets from the union of these four profiles (0.92, [0.90, 0.93]) is the same as that obtained for the set from the union of all simultaneous bivariate profiles. This is achieved at approximately 19% of the computational cost (460 thousand vs 2.36 million likelihood calls, Appendix K.2, Table 14). With the current settings, this costs around 460 thousand likelihood calls on average ($122 + 90 + 135 + 111 = 458$ thousand, Appendix K.2, Table 14), which is significantly less than the ten million calls the full trajectory confidence set required to hit effectively the same level of coverage (0.90 [0.84, 0.93], Section 5.1.2).

This demonstrates that selecting a small subset of simultaneous bivariate profiles with coverage close to the union of all these profiles is generally possible. Additionally, it demonstrates the value of this profile-wise methodology. Namely, the statistical performance of the profile-wise prediction sets is as good, or better, on this model than prediction sets from the naive sampling approach while having a significant computational performance advantage (i.e. 4.6% of the total cost of the sampling approach under the current settings *and* the sampled set used better-informed parameter bounds).

To get high levels of reference tolerance set coverage, we need to include profiles that contain σ in our subset as this is the most important parameter for quantifying uncertainty in the $1 - \delta$ population reference set (Section 6.3.2). We try to maximise efficiency by concurrently meeting the model trajectory and reference set coverage goals. Resultantly, we begin with the four profiles used for the trajectory confidence set. We then include the bivariate profile, which produces the profile-wise reference tolerance set with the highest simultaneous coverage in Figure 43a, $\psi \in \{(\lambda_2, \sigma)\}$.

The coverage of the profile-wise reference tolerance set from the original four profiles is 0.88 [0.85, 0.90] (Figure 43a), or slightly lower than the coverage of its profile-wise trajectory confidence set (0.92 [0.90, 0.93]). This means that, on average, the value of σ in the nuisance parameters for each point in these parameter confidence sets was lower than the true value (Section 5.1.2). When we then include the fifth profile, (λ_2, σ) , in this union, the coverage of the profile-wise reference tolerance set improves to a comparable level (0.91 [0.89, 0.93], Figure 46b) to the union's coverage of the model trajectory. These subsets of four and five simultaneous bivariate profiles produce profile-wise reference tolerance sets with improved coverage of about 0.1 over the sets from random subsets of four and five profiles, respectively. This is smaller than the difference observed for the profile-wise trajectory confidence set but remains a good performance.

6.3.4.3 Potential Methods For Finding Good Subsets

There are several potential methods for finding good subsets of profiles, some of which we have

already covered. This is by no means an exhaustive list. Nonetheless, it remains a good starting point for further research. Firstly, by considering the pointwise coverage of the relevant prediction set from simultaneous bivariate profiles as used in Section 6.3.4. Secondly, to reduce the computational cost of the coverage simulation, we could instead consider the same pointwise coverage but for simultaneous univariate profiles. We would then choose bivariate profiles from pairwise combinations of univariate profiles with complementary pointwise coverage properties (Section 6.3.4.1). Thirdly, we could consider the dominant parameters of the model equations either using asymptotic analysis or the previous pointwise approach (Section 6.3.1.1). Fourthly, if we have multiple observed model components that are weakly coupled, we create a subset, from each component, that contains parameter combinations that are primarily featured in the equation of that component (Section 6.3.4.2). Finally, an approach like PCA [118] may be viable by considering the eigenvalues and eigenvectors of the observed FIM, $\mathcal{H}(\hat{\theta})$, ideally in a logged parameterisation [50] (approach taken in ‘sloppiness’ literature).

In the PCA approach, after the eigenvalue decomposition of $\mathcal{H}(\hat{\theta})$ we choose a subset by considering the n most stiff eigenvector directions which correspond to the n largest eigenvalues [50]. Parameters to use in profiles are given by the corresponding elements of these stiff eigenvectors with the largest magnitudes. For example, say the stiffer eigenvector in a three-parameter model is $[0, 1, 0]$, then we would choose to profile the second parameter. The primary idea is that the model is most sensitive to parameter changes along the stiffer directions [50]. Therefore, uncertainty in these stiffer directions should impact the uncertainty of predictive quantities to the greatest degree. However, because $\mathcal{H}(\hat{\theta})$ is a local approximation, this approach may give inaccurate results where nonlinear interactions are involved.

6.4 Performance Against Full Parameter Confidence Sets

Here, we compare the simultaneous coverage performance on each of the profile-wise prediction sets considered in this section and Section 5 with the prediction sets from full parameter confidence sets in Section 5.1. To complete this comparison, we will consider the computational cost required to obtain each coverage result in terms of the number of likelihood function calls. This allows us to quantitatively demonstrate the value of the PWA workflow in terms of statistical and computational performance. Moreover, it highlights the ways in which the computational performance advantage of the PWA workflow can be further increased. The mean number of likelihood function calls of each of the profile-based sets, including specification of the settings used, can be found in Appendix K.

The mean number of likelihood evaluations and the number of samples taken are equivalent for these comparisons. These values define the computational cost of finding each parameter confidence set.

However, there are additional costs of propagating these sets forward into predictive quantities.

For the trajectory confidence set, we have to evaluate the model for each point in the parameter set at the desired timepoints. Similarly, for the reference tolerance set, we first evaluate the trajectory confidence set and then evaluate the associated reference intervals at the desired timepoints. Resultantly, the actual ‘model evaluation’ cost should include around another 1000 calls for each of the full parameter confidence sets with coverage of around 0.95 and $30 \text{ or } 50 \times |\theta|$ for each of the bivariate sets. This cost is small in comparison to the actual sampling of the full parameter confidence set (one thousand vs 30 thousand or 500 thousand or ten million) or the bivariate sets (e.g. 150 vs 54700, Appendix K.2, Table 12). Therefore, it will be ignored. Similarly, evaluating each reference interval is very performant as it involves an efficient 1D optimisation across a univariate, unimodal distribution [5]. These evaluations are not tied to the model evaluation cost, so they are also ignored.

Furthermore, we ignore the cost of finding $\hat{\theta}$, which is low relative to the cost of finding bivariate boundary points. We also ignore the cost of evaluating the hessian at $\hat{\theta}$, $\mathcal{H}(\hat{\theta})$, and subsequent calculations derived from this matrix, such as the calculation of Wald confidence intervals. For these small models, these incur a very low cost.

The computational performance of the sampled full parameter confidence sets relative to the coverage performance is impacted by the parameter bounds used. Resultantly, the performance of these methods could be improved with more well-informed parameter bounds. However, the inverse of this is also true. In particular, we regard the parameter bounds chosen for sampling from the two-species model as quite well-informed, as discussed in Section 5.1.1. Therefore, we regard this as a relatively fair comparison with our profile-wise methods, which do not have a significant performance dependence on parameter bounds (for bivariate profiles, see Section 4.3.3), allowing them to use less well-informed parameter bounds.

6.4.1 Improving Computational Performance of Profile-Wise Prediction Sets

The computational performance of our profile-wise methods could also be improved in at least eight ways, as discussed throughout this text. In particular, we could:

1. further increase the absolute tolerance value in `find_zero`, $\log_{10}(\text{atol}) > -3$. However, this would reduce the correctness of the boundaries found; it is not strictly recommended to modify this past its current value.
2. using better-informed guesses for nuisance parameter values in the optimisation scheme. However, this could result in convergence issues, as noticed in Section 3.3 for `LikelihoodPro-`

- `filer`'s linear extrapolation method.
3. removing duplicate optimisation function calls when evaluating univariate confidence intervals and points on the bivariate boundary by implementing a bespoke version of Brent's method. This could reduce the number of optimisation calls (and similarly likelihood calls) by around 25% when finding parameter confidence intervals (Section 3.1.3.5) and around 30% when finding bivariate boundaries (Section 4.3.1).
 4. adapting the `LikelihoodProfiler` CICO approach to work in bivariate interest parameter space. Its performance is shown to be significantly better than our simple bracketing methodology when calculating parameter confidence intervals in Section 3.3, particularly for higher numbers of parameters.
 5. adjust optimisation scheme settings that have not been considered. For example, using a derivative-based algorithm in conjunction with an automatic differentiation scheme. `LikelihoodBasedProfileWiseAnalysis` is set up to allow this to be considered very easily.
 6. further increase the parameter relative tolerance in the optimisation scheme, $\log_{10}(\text{xtol_rel}) > -12$. This will enable faster convergence to local maxima but potentially result in less accurate solutions. Errors in `find_zero` might also occur if the optimisation scheme is not able to correctly evaluate the sign of $f(\psi)$ (Section 3.1).
 7. reduce the number of boundary points found in bivariate profiles from the 50 and 30 currently used; we may only need around 10 or 20, as long as they have good coverage of the boundary, which will be the case for the `IterativeBoundaryMethod` on the models considered, which have reasonably convex bivariate boundaries.
 8. reduce the number of profile combinations considered, as discussed in Section 6.3.4.

The first four of these require a significant change in the implementation or are not recommended to be modified beyond their current value, so they will not be considered. The fifth approach involves adjusting other optimisation scheme settings, which we will also not consider here. However, we note that these other optimisation settings could significantly improve the performance of the profile-wise methodology, e.g. if a more appropriate optimisation algorithm were used for a given model. Testing the last three approaches is straightforward and will be considered here for prediction sets from simultaneous bivariate profiles. These approaches could also be applied to the profile-wise prediction sets from the other profiles; a similar computational performance improvement would be observed.

In particular, the subset for the reduced number of simultaneous bivariate profile combinations is

taken from the subsets in Section 6.3.4. Additionally, we set $\log_{10}(\text{xtol_rel}) = -8$, from -12 . We reduce the number of boundary points to 20 in the logistic and Lotka-Volterra models and 10 in the two-species logistic model. These two setting adjustments are the ‘less conservative’ settings. We expect to observe a significant reduction in computational cost when using either a reduced number of profile combinations, the less conservative settings or the combination of the two.

6.4.2 Logistic and Lotka-Volterra Models

The performance of each of the approximate parameter confidence sets considered in this text when propagated into trajectory confidence sets for the logistic and Lotka-Volterra models can be seen in Figure 47. To keep things fair, we do not include the full parameter confidence set sampled from the original restrictive bounds in this comparison. Univariate and bivariate profiles labelled as ‘profile path’ are simultaneous univariate and simultaneous bivariate profiles, respectively.

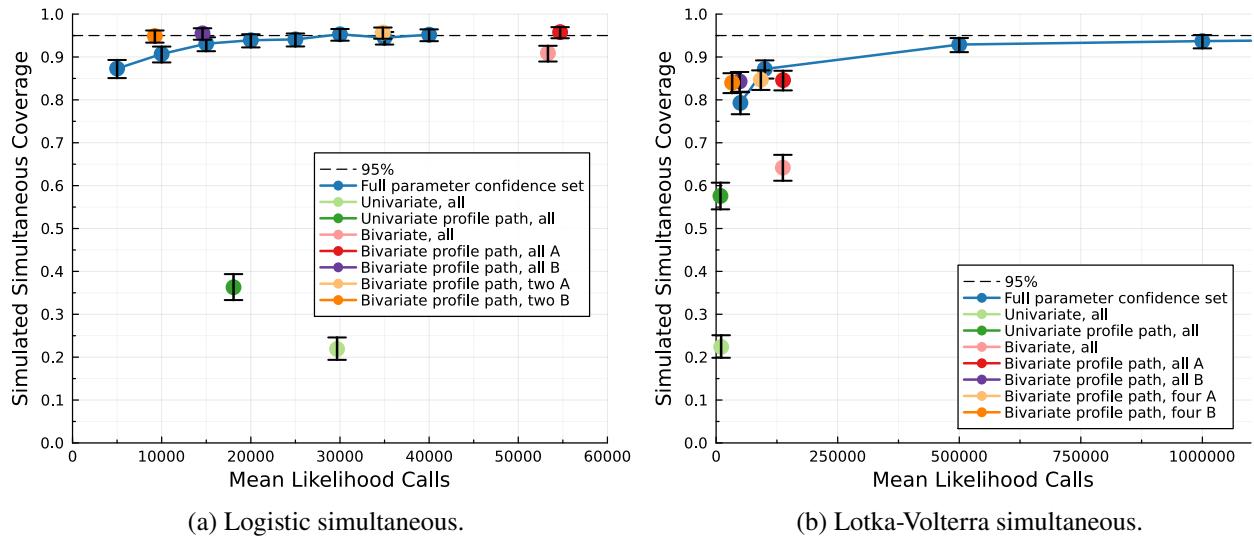


Figure 47: Mean number of likelihood function calls versus the coverage of trajectory confidence sets in the logistic and Lotka-Volterra models [1] propagated forward from each of the parameter confidence sets considered. ‘A’ at the end of a legend entry means the original settings are used. ‘B’ at the end of a legend entry means the less conservative settings are used. The two simultaneous bivariate profiles for the logistic model are $\psi \in \{(\lambda, K), (\lambda, C(0))\}$ (Section 6.3.4). 95% confidence intervals for the simulated coverage are provided as error bars.

Here, we see the poor simultaneous coverage performance of profile-wise trajectory confidence sets from the union of univariate profiles at the univariate and simultaneous asymptotic thresholds. The coverage results are significantly lower than the 0.95 obtained by the full trajectory confidence set. The computational cost of the univariate profiles is relatively high on the logistic model because these profiles also include 20 points between each interval. Additionally, the cost difference between this model’s two univariate parameter sets is expected to be predominantly random noise, given that

only 100 simulations were used to evaluate this mean. Nonetheless, the coverage performance of the trajectory confidence set from the union of simultaneous univariate profiles in the Lotka-Volterra model is not bad, given the only 8910 likelihood calls required (Appendix K.1, Table 10). Additionally, these profiles can be used to observe the importance of each parameter for determining uncertainty in the model trajectory at different times. This indicates that univariate profiles can still be useful for the initial evaluation of uncertainty in the model trajectory at a relatively low cost.

Meanwhile, the profile-wise trajectory confidence set from the union of bivariate profiles shows a significant improvement in simultaneous coverage performance over the univariate profiles. The coverage of profile-wise trajectory confidence sets from bivariate profiles formed at the bivariate asymptotic threshold, $\ell_c = -\Delta_{2, 0.95}/2$, still exceeded the coverage of sets from the union of simultaneous univariate profiles in both models despite the simultaneous threshold, $\ell_c = -\Delta_{|\theta|, 0.95}/2$, $|\theta| \geq 3$, being larger than the bivariate threshold. This is likely due to the incorporation of the interaction between two parameters in the bivariate profiles.

The profile-wise trajectory confidence sets from the union of simultaneous bivariate profiles reach the desired coverage of 0.95 on the logistic model. This matches the full trajectory confidence set's coverage while requiring about 80% additional calls using the original settings ($100\% \times ((54700 - 30000)/30000) \approx 82\%$, Appendix K.2, Table 12). Using the less conservative settings reduces the computational cost of these simultaneous bivariate profiles to 14600 calls (Appendix K.2, Table 12) with no coverage penalty. This is around a 50% decrease in computational cost relative to the full trajectory confidence set ($100\% \times ((14600 - 30000)/30000) \approx -51\%$). If we then choose the subset of two simultaneous bivariate profiles (Section 6.3.4.1) with the less conservative settings, this cost reduces by an additional 30% to around 9240 calls ($5780 + 3460 = 9240$, Appendix K.2, Table 12), or a third of the cost for the same coverage. These points are in the top left quadrant of the figure, above the blue line for the full parameter confidence set: this is the sign of a successful use of the PWA workflow.

The profile-wise trajectory confidence sets from the union of simultaneous bivariate profiles show their approximate nature on the Lotka-Volterra model. The sets fail to reach the coverage of the full trajectory confidence set (only 0.85 [0.82, 0.87] instead of 0.95 [0.93, 0.96]). At the original settings, which required a mean of 137000 likelihood calls (Appendix K.2, Table 13), the simultaneous coverage of its profile-wise trajectory confidence sets lies slightly below the line of the coverage observed between 100000 and 500000 samples. When instead using the less conservative settings, the computational cost is reduced by around 65% ($100\% \times ((48200 - 137000)/137000) \approx -65\%$, Appendix K.2, Table 13)

for no decrease in coverage (to two d.p.). This is more than the 0.79 [0.77, 0.82] simultaneous coverage mustered by the full trajectory confidence set using more than 50000 likelihood calls and only slightly less than the 0.87 [0.85, 0.89] mustered for twice as many likelihood calls ($48200 \times 2 < 100000$). Moreover, if we then choose the subset of four simultaneous bivariate profiles (Section 6.3.4.1) with the less conservative settings, this cost reduces a further 32% ($100\% \times ((33000 - 48200)/48200) \approx -32\%$; $7800 + 8470 + 8410 + 8270 \approx 33000$, Appendix K.2, Table 13), for only a slight reduction in the simultaneous coverage of the corresponding profile-wise trajectory confidence set (0.84 [0.82, 0.86]). These points are again in the top left quadrant of the figure and above the blue line; the PWA workflow has again been successful at obtaining a good coverage result that is better than a sampled approach at the same level of computational cost. However, this result does underline the approximate nature of the PWA workflow.

6.4.3 Two-Species Logistic Model

The improvement in computational cost for the same or better simultaneous coverage of predictive quantities is seen to an even greater degree when considering the two-species logistic model in Figure 48. We now see many points present in the top left quadrant above the blue line, with a significant difference between the number of likelihood calls required to reach the same level of coverage, and similarly the coverage at the same number of likelihood calls. Again, univariate and bivariate profiles labelled as ‘profile path’ are simultaneous univariate and simultaneous bivariate profiles, respectively.

The profile-wise prediction sets from the union of all univariate profiles again show poor simultaneous coverage performance. Meanwhile, the profile-wise reference tolerance set from the union simultaneous univariate profiles has a significant improvement in coverage to 0.83 [0.80, 0.85] for a slight cost in cost in performance (56900 instead of 49200, Appendix K.1, Table 11). This is a similar level of coverage to the full reference tolerance set formed using two million samples while using 35 times fewer evaluations ($2000000/56900 \approx 35.1$). This difference in computational cost could be made even more stark if the less conservative settings were used. Additionally, it would be even larger if the sampled set used the less well-informed parameter bounds utilised by the profiles (Appendix A.3.1).

Meanwhile, profile-wise trajectory confidence sets from the union of bivariate profiles again show a significant improvement in simultaneous coverage over the univariate profiles (0.57 [0.54, 0.60] instead of 0.06 [0.04, 0.07] and 0.43 [0.40, 0.46]). This is despite the considerable difference between the bivariate threshold, $\ell_c = -\Delta_{2,0.95}/2 \approx -3$, and the simultaneous threshold used for the simultaneous univariate profiles $\ell_c = -\Delta_{7,0.95}/2 \approx -7$. This again underlines the importance of evaluating the interactions between (at least) two parameters when simultaneously predicting the model trajectory.

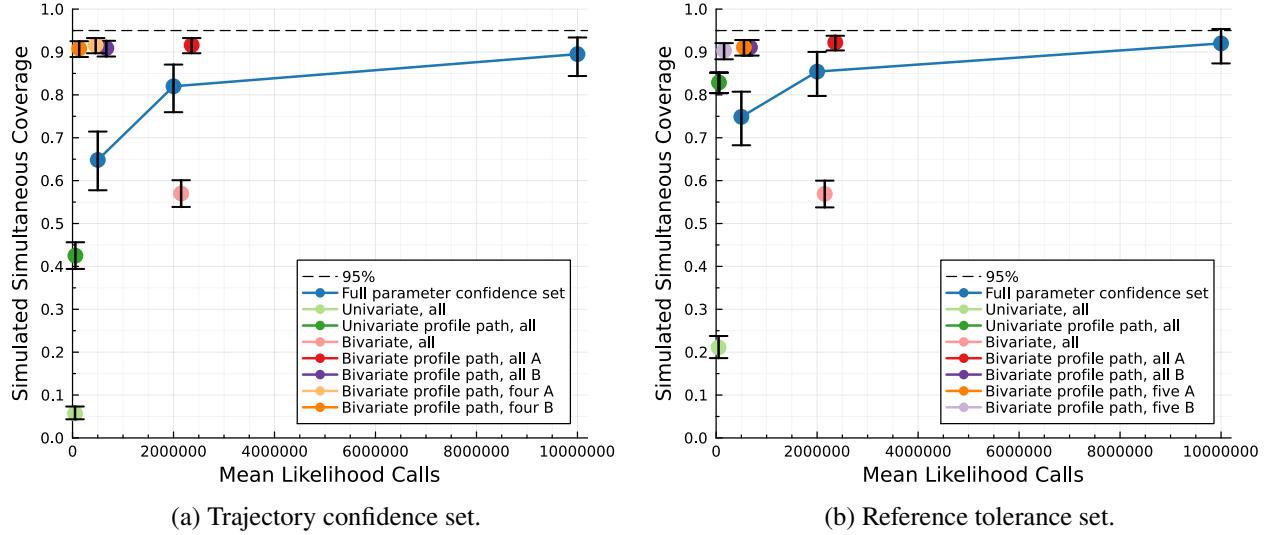


Figure 48: Mean number of likelihood function calls versus the coverage of trajectory confidence sets and reference tolerance sets in the two-species logistic model [2] propagated forward from each of the parameter confidence sets considered. ‘A’ at the end of a legend entry means the original settings are used. ‘B’ at the end of a legend entry means the less conservative settings are used. The four bivariate profiles are $\psi \in \{(\lambda_1, K), (\lambda_1, C_1(0)), (\lambda_2, \delta), (\lambda_2, C_2(0))\}$ (Section 6.3.4). The five bivariate profiles include the previous four, with the addition of $\psi \in \{(\lambda_2, \sigma)\}$. 95% confidence intervals for the simulated coverage are provided as error bars.

The profile-wise prediction sets from the union of simultaneous bivariate profiles have simultaneous coverage point estimates higher or the same as those for the prediction sets from the full parameter confidence set that uses ten million likelihood calls (0.92 [0.90, 0.93] vs 0.90 [0.84, 0.93] and 0.92 [0.90, 0.94] vs 0.92 [0.87, 0.95]). At the current settings, this is accomplished at a 76% decrease in computational cost relative to the full parameter confidence set ($100\% \times ((2.36 - 10) - 10) \approx -76\%$, Appendix K.2, Table 14)). If we then use the less conservative settings on these profiles, we reduce this cost by a further 72% ($100\% \times ((0.669 - 2.36) - 2.36) \approx -72\%$) or a decrease of 93% from the cost of the full parameter confidence set ($100\% \times ((0.669 - 10) - 10) \approx -93\%$). This is a drastic computational performance improvement while maintaining the desired coverage properties of these profile-wise prediction sets.

We can take this even further; when considering just the four profiles for the profile-wise trajectory confidence set (Section 6.3.4.2), using the less conservative settings reduces the number of likelihood calls to 132000 ($32500 + 26300 + 40900 + 32300 = 132000$, Appendix K.2, Table 14). This is barely 1.3% of the original ten million calls. Similarly, if we consider the five profiles for the profile-wise reference tolerance set (Section 6.3.4.2), using the less conservative settings reduces the number of likelihood calls to 158000 ($132000 + 25900 = 158000$, Appendix K.2, Table 14) or barely 1.6% of the original ten million calls for a slight drop in coverage (0.90 [0.88, 0.92]). Again, these

computational differences would be even larger if the sampled full parameter confidence set used the less well-informed parameter bounds utilised by the profiles (Appendix A.3.1).

7 Discussion and Conclusions

In this section, we review the main findings of this report, including their limitations, while considering the research objectives laid out in Section 1.1. We will discuss opportunities for future work based on these findings and research objectives.

7.1 Univariate Profiles

We provide a straightforward step-wise approach based on a bracketing method for evaluating univariate confidence intervals, as implemented in `LikelihoodBasedProfileWiseAnalysis`, with Wald confidence intervals used to improve the efficiency of this approach. This method is appropriate for univariate profile log-likelihood functions that are unimodal and continuous. These Wald confidence intervals may be unsuccessful in improving efficiency or even create a higher computational overhead if the likelihood function is not sufficiently regular, e.g. if the hessian cannot be computed or is singular. Nonetheless, their use is recommended on identifiable models as they decreased the number of likelihood calls by around 11% on the logistic model considered (Appendix A.1).

7.1.1 Ellipse Approximations

We show that, for a sufficiently regular likelihood, the analytical ellipse approximation of the profile log-likelihood function and the full ellipse approximation equation, which requires the optimisation of nuisance parameters, produce consistent profiles when parameters are unconstrained. This directly connects the analytical method for finding Wald confidence intervals with the method used for the profile log-likelihood function. The Wald confidence interval from the full ellipse approximation may be more appropriate for propagation forward into prediction space when parameter feasibility constraints need to be enforced.

7.1.2 Implementation Validation

We validate the implementation by exploring the frequentist coverage properties of 95% confidence intervals for parameters. For a sufficient number of observations, we find that the coverage of each parameter confidence interval is approximately the desired value of 0.95 in the models considered (Appendices A.1, A.2, A.3 and A.3.1). Convergence issues are observed in the two-species logistic model with a Gaussian data distribution. We instead suggest using a logit-normal data distribution which resolves these convergence issues, without changing confidence interval coverage.

The coverage tests are shown to serve as a useful diagnostic tool for indicating whether a specified number of observations, at a particular quality level, is sufficient for obtaining the desired coverage of parameter confidence intervals. Therefore, the test could be used prior to data collection to quantify

the minimum amount of data to collect. Similarly, it could be used after data collection to indicate whether the quality and amount of data collected is sufficient for obtaining the desired coverage.

The implementation of a bootstrap or Monte-Carlo based procedure for calibrating the asymptotic threshold, ℓ_c , may be useful future work, to enable correct coverage of parameter confidence intervals when there is not a sufficient number of observations, e.g. Warne, Baker, and Simpson [119]. For example, this would be useful for the two-species logistic model evaluated in this work; performing a larger number of coral coverage surveys may be highly unrealistic due to the time and resources required.

7.1.3 Implementation Performance Improvements and Comparisons

The implementation of our confidence interval method can be improved by implementing a bespoke bracketing method. This would reduce the number of duplicate optimisation function calls and thus computational cost by around 25%. This is a recommended future update to `LikelihoodBasedProfileWiseAnalysis`; these cost savings would also apply to the methods used for bivariate profile boundaries.

In comparison to `LikelihoodProfiler`'s CICO method [22], our method for finding parameter confidence intervals, in the three models considered, required on average 39.7%, 66.5% and 168% more likelihood calls, respectively. The performance gap widens as more parameters are considered; the three models have three, four and seven parameters, respectively. It demonstrates that the CICO method is more appropriate for finding parameter confidence intervals. Therefore, if only these intervals are desired, then `LikelihoodProfiler` should be used instead of `LikelihoodBasedProfileWiseAnalysis`. Consequently, switching to the CICO approach could improve the performance of `LikelihoodBasedProfileWiseAnalysis`. Using Wald confidence intervals to reduce the search bracket of the CICO method could be useful future work to improve the performance of their method. However, the additional complexity may be less valuable in their method, given that it could increase computational overhead for less regular likelihoods.

7.2 Bivariate Profiles

Using pseudocode, we describe the `Fix1Axis` method for finding bivariate profile boundaries produced by Simpson and Maclaren [1]. Five issues are identified with this method that are improved on using an iterative process in which we propose six alternative methods (math-heuristics), as implemented in `LikelihoodBasedProfileWiseAnalysis`. We use a bracketing method approach to find each boundary point, as used for the method for univariate confidence intervals, so areas for improvements

to that method also apply here. We also provide a simple procedure for efficiently sampling points inside a bivariate profile's boundary once the boundary has been found.

7.2.1 Performance of Boundary Methods

Of the developed methods, `RadialMLEMethod` and `IterativeBoundaryMethod` had the best computational performance. When finding a desired number of boundary points on each of the three bivariate profiles in the logistic example, these methods required around ten optimisation calls per boundary point, with the next best method requiring 14. Additionally, in contrast to the other methods, the computational performance of these methods is relatively unaffected by the width of parameter bounds.

These two methods also had the best area coverage of the true bivariate boundary for a specified number of boundary points. This means these two methods are the best at quantifying uncertainty in the bivariate profile space. However, the `IterativeBoundaryMethod` is better than the `RadialMLEMethod` for concave boundaries because it can identify locations on the boundary polygon that need to be explored. As a result, it is used for all subsequent tests of the PWA workflow. Area coverage results for the `IterativeBoundaryMethod` are used to choose the number of boundary points in subsequent testing on the three models considered (50, 30 and 30, respectively).

Investigation of other methods for finding the boundary of bivariate profiles could be valuable future work for further improving the computational performance of the PWA workflow. For example, a Bayesian LSE approach or application of the `InformationGeometry` approach [23] in 2D interest parameter space.

On the logistic model (Appendix A.1), the `IterativeBoundaryMethod` is shown to be qualitatively more appropriate for sampling points of a bivariate profile than a rejection sampling approach while being unaffected by the impact of parameter bounds. Additionally, even when parameter bounds are very well informed, the `IterativeBoundaryMethod` in conjunction with a method to sample inside the boundary polygon is more computationally efficient for large numbers of points (sampling internal points has a cost of just over one optimisation call per point). Moreover, as found in Sections 5 and 6, boundary points are more valuable than internal points for forming profile-wise prediction sets, so the value of each point in the sets found is not equivalent. This further reinforces the value of using the `IterativeBoundaryMethod` over a rejection sampling approach.

7.2.2 Implementation Validation

We again validate the implementation by exploring the frequentist coverage properties of 95% confidence boundaries for bivariate interest parameters. For a sufficient number of observations, we find that the coverage of each bivariate boundary was approximately the desired value of 0.95 in the models considered (Appendices A.1, A.2, and A.3.1). We observe similar coverage issues for the two-species logistic model with a logit-normal data distribution to those seen for parameter confidence intervals. These issues indicate that coverage in the confidence interval space is directly correlated with the coverage of the interest parameter pair in the bivariate profile boundary space.

7.3 Profile-wise Analysis

7.3.1 Prediction Sets From Full Parameter Confidence Sets

The coverage of trajectory confidence sets and reference tolerance sets from propagating forward full parameter confidence sets at a 95% confidence level can reach approximately 0.95 for a sufficiently large number of sampled points on the three considered models. This level of coverage is used to demonstrate how well the approximate profile-wise prediction sets perform.

When the standard deviation of the data distribution, θ^0 , is estimated, we find that the coverage of full reference tolerance sets is generally similar to that of full trajectory confidence sets, although slightly different (slightly lower or higher).

We demonstrate that simultaneous coverage of individual observations with 95% confidence is difficult when using full (95%, 95%) reference tolerance sets. Each model's best full reference tolerance set failed to reach simultaneous coverage of more than 0.75 [0.68, 0.81]. This was unsurprising, given that a (100%, 95%) reference tolerance set is technically required to meet this goal when the number of observations goes to infinity.

In contrast, the alternative goal, to contain 95% of all observations with 95% confidence, is very well covered by the full (95%, 95%) reference tolerance sets. These sets had over 0.98 of this alternative goal on each model, even with increasing $|\theta|$. Therefore, we recommend consideration of this alternative goal over capturing 100% of observations simultaneously. Moreover, the alternative goal aligns elegantly with the defined reference tolerance sets, underlining the additional value of introducing reference tolerance sets within the PWA workflow.

7.3.2 Profile-Wise Prediction Sets From Univariate and Bivariate Profiles

The key finding for profile-wise prediction sets from the union of all univariate or bivariate profiles is that setting the asymptotic threshold, ℓ_c , based on the dimensionality of the profile's interest parameter

has a large impact on the observed coverage performance. This is because the extremities of the profile paths do not touch the extremities of the full parameter confidence set for $|\theta| > 1$ (univariate) and $|\theta| > 2$ (bivariate). Additionally, these extremities get further apart for increasing $|\theta|$. Several cases were noticed where evaluating the simultaneous asymptotic threshold with a confidence level equal to the observed coverage of a profile-wise prediction set would produce a threshold similar to the corresponding 95% confidence univariate or bivariate threshold. Furthermore, the performance of profile-wise prediction sets from either profile dimension worsened as $|\theta|$ increased in the three considered models. Therefore, using the simultaneous asymptotic threshold for profiles may be more appropriate for forming profile-wise prediction sets, at least in identifiable models.

The univariate profile for the standard deviation of the data distribution, σ , provided no value in predicting the model trajectory. This is because all nuisance parameters along the profile are set to their corresponding MLE value, $\hat{\theta}^M$. In contrast, this univariate profile was the most important for predicting the $1 - \delta$ population reference set, as it directly defined the width of the reference interval. The profile-wise reference tolerance set from σ had simultaneous coverage of 0.09 [0.07, 0.11], while the sets from all other univariate profiles had coverage of effectively 0.0. In particular, the upper endpoint of the confidence interval for σ provides all uncertainty quantification for this parameter with a Gaussian or logit-normal data distribution.

Evaluating the non-boundary points of univariate and bivariate profiles does not increase the coverage of profile-wise prediction sets from these profiles, except for a single case. The coverage of profile-wise trajectory confidence sets from univariate profiles in the logistic model benefits from internal sampling points. This means that the extremities of the univariate profiles (generally) and bivariate profiles define the extremities of the profile-wise prediction sets, at least on the models considered. Therefore, assuming that the boundaries of the bivariate profiles map to the extremities of the profile-wise trajectory confidence sets was an appropriate assumption in Simpson and Maclaren [1].

As expected, profile-wise prediction sets from the union of bivariate profiles have improved simultaneous coverage properties over those from the union of univariate profiles, primarily due to improved pointwise coverage. For profile-wise trajectory confidence sets, this improves from 0.22 [0.19, 0.25], 0.23 [0.21, 0.26] and 0.06 [0.04, 0.07] to 0.91 [0.89, 0.93], 0.64 [0.61, 0.67] and 0.57 [0.54, 0.60], respectively on the three, four and seven parameter models considered. For the profile-wise reference tolerance set, this improves from 0.21 [0.19, 0.24] to 0.57 [0.54, 0.60] on the seven-parameter model. The improvement in pointwise coverage is most significantly down to bivariate profiles capturing the interaction between two parameters.

We also observe that the profiles of particular parameters are most important for capturing uncertainty at particular points in time. For example, the parameter K in the logistic model, which is the carrying capacity density, produces a profile-wise trajectory confidence set with pointwise coverage > 0.9 at late time, but ≈ 0.4 otherwise.

The coverage of the alternative observation goal is above 0.95 for the profile-wise (95%, 95%) reference tolerance sets from the union of univariate profiles for the three and four parameter models (0.97 [0.96, 0.98] and 0.995 [0.99, 1.0]). However, it is lower than desired for the seven-parameter model at 0.72 [0.70, 0.75]. Therefore, profile-wise reference tolerance sets from the union of univariate profiles could be useful for this particular metric. However, resolving coverage issues related to $|\theta|$ is required to observe the desired coverage on higher dimensional models.

Similarly, the coverage of the alternative observation goal is above 0.95 for the profile-wise (95%, 95%) reference tolerance sets from the union of bivariate profiles for the three and four parameter models (approximately 1.0). In the seven-parameter model, this drops to 0.93 [0.91, 0.94]. This further indicates that the asymptotic threshold constrains this coverage as $|\theta|$ increases.

7.4 Profile Paths For Profile-Wise Analysis

We define a new approach within the PWA workflow called *profile paths* for profile-wise analysis. This approach uses profiles evaluated at the simultaneous asymptotic threshold to form profile-wise prediction sets. This is to resolve issues with the coverage of these sets related to setting the degrees of freedom of the threshold, ν , equal to the interest parameter dimensionality, $|\psi|$. This new approach means that when evaluating the boundary of a simultaneous bivariate profile, we evaluate good ‘paths’ of points along the extremities of the full parameter confidence set.

The primary limitation of our evaluation of the profile path approach is that we only consider models which are relatively well-identified at the simultaneous asymptotic threshold. If the model is not well-identified, then it is probable that setting $\nu = |\theta|$ is not appropriate, as the degrees of freedom in parameter space may be lower than $|\theta|$. We recommend applying the profile paths approach to the models discussed in Simpson and Maclaren [16] to explore its validity in this context. Another appropriate model may be the STAT5 model from Borisov and Metelkin [22] and Boehm et al. [116], which also has more parameters than the models discussed here (nine).

7.4.1 Profile-Wise Prediction Sets From Simultaneous Univariate Profiles

The profile-wise trajectory confidence sets from the union of simultaneous univariate profiles show improved simultaneous coverage of the model trajectory, and the 95% population reference set when

the data distribution standard deviation is known, over the sets from univariate profiles. The coverage performance is similar on each model (0.36 [0.33, 0.39], 0.58 [0.54, 0.61] and 0.43 [0.40, 0.46], respectively). The simultaneous coverage of the set from each profile is still effectively 0.0; only the union of profiles is useful for simultaneous coverage of the model trajectory in the considered models. However, this is still worse than the simultaneous coverage of these sets formed by the union of bivariate profiles. This demonstrates the importance of the interaction between two parameters when making simultaneous predictions for the model trajectory.

Sampling points within the confidence interval for each parameter is useful for simultaneous coverage of the model trajectory on only the logistic model. This is likely because points along the profile which improve coverage are close to boundary points in full parameter space that are as good or better than these points. This explanation is supported by the findings for internal points of bivariate profiles on the models considered.

Pointwise coverage is above 0.95 at least 60% of the time for the profile-wise trajectory confidence sets from the union of simultaneous univariate profiles in the models considered; particularly for the higher dimensional models, the pointwise coverage is reasonable. Resultantly, even though simultaneous coverage is lower than desired, evaluating profile-wise trajectory confidence sets from the union of simultaneous univariate profiles can still capture a large amount of uncertainty in predictive quantities. Moreover, this can be done relatively cheaply compared to bivariate profiles.

The pointwise coverage plots of these profile-wise trajectory confidence sets also provide useful sensitivity information. When the pointwise coverage of the set from the union of simultaneous univariate profiles is high, the uncertainty in the model trajectory is effectively dependent on the uncertainty of a single dominant parameter; the trajectory effectively has a one-dimensional approximation. Similarly, when dips in coverage occur, they can be explained as times when the model trajectory is dependent on multiple parameters. Often, this occurs when the trajectory transitions from being dominated by a particular parameter to another parameter (or sets of parameters to another set of parameters). Therefore, considering the interaction between at least two parameters is valuable for predicting the model trajectory.

The simultaneous coverage of the profile-wise (95%, 95%) reference tolerance sets from the union of simultaneous univariate profiles is significantly improved over the coverage of the model trajectory when the data distribution standard deviation is estimated. We observed simultaneous coverage of 0.83 [0.80, 0.75], up from the model trajectory result of 0.43 [0.40, 0.46]. This is a significant improvement on the reference tolerance sets from the union of univariate profiles, which only managed 0.21 [0.19,

0.24]. The reference sets from the upper endpoint of the profile for σ have simultaneous coverage of 0.61 [0.57, 0.64]. This parameter is the most important parameter for quantifying uncertainty in the $1 - \delta$ population reference set while being the least important for the model trajectory set. Therefore, the evaluation of profiles which include this parameter is dependent on whether uncertainty quantification of observations (the population reference set) is regarded as important.

The simultaneous coverage of our alternative observation goal for the profile-wise (95%, 95%) reference tolerance sets from the union of simultaneous univariate profiles on the models considered. This indicates that if capturing (e.g. 95%) of all observations is your goal, then forming approximate (95%, 95%) reference tolerance sets using the union of simultaneous univariate profiles will provide excellent performance at a very low computational cost.

7.4.2 Profile-Wise Prediction Sets From Simultaneous Bivariate Profiles

The simultaneous coverage of profile-wise trajectory confidence sets from the union of simultaneous bivariate profiles is either very close to or reaches the same level of coverage as the full trajectory confidence set it was approximating (0.95 [0.94, 0.96] vs 0.95 [0.94, 0.96], 0.85 [0.82, 0.87] vs 0.95 [0.93, 0.96] and 0.92, [0.90, 0.93] vs 0.90 [0.84, 0.93]). The reduction in simultaneous coverage observed for bivariate profiles with increasing $|\theta|$ is not present. These results emphasise these profile-wise sets' approximate nature, which converges to the coverage of full trajectory confidence sets as $|\psi|$ increases. Nonetheless, the coverage performance is reasonable in all cases, as the pointwise coverage for the set from the union of simultaneous bivariate profiles is typically at least 0.95. We also find we can improve in cases where simultaneous coverage is imperfect by sampling features in the data that are highly dependent on three or more parameters.

The simultaneous coverage of the profile-wise (95%, 95%) reference tolerance set (0.92 [0.90, 0.94]) from the union of simultaneous bivariate profiles is *as good* as the coverage observed from the full parameter confidence set from ten million samples (0.92 [0.87, 0.95]). This is much improved from the 0.57 [0.54, 0.60] observed for the set from the union of bivariate profiles formed at the bivariate. This demonstrates that setting $\nu = |\theta|$ for these profiles is sensible for this model.

The profile-wise (95%, 95%) reference tolerance sets from each simultaneous bivariate profile have coverage very close to 1.0 for the three and four parameter models, while they are closer to 0.9 for the seven parameter model. The sets from the union of simultaneous bivariate profiles have coverage of this goal of effectively 1.0.

We find that only boundary points of the simultaneous bivariate profiles are important for capturing uncertainty in predictive quantities. This finding is limited by the number of models considered; it

may only hold for some models. Nonetheless, in models where this is the case, other methods like random sampling or MCMC may be wasting computational time evaluating points not located near or on this boundary. Our method can target these points, which provides a clear performance advantage in these models.

Therefore, to achieve better coverage performance of predictive quantities when using the profile-wise approach, we recommend setting the degrees of freedom used for the asymptotic threshold equal to the number of model parameters. This recommendation is limited by the models we have considered; different results may be observed for models with very high numbers of parameters (e.g. 10-20+) or non-identifiable models. Therefore, we recommend repeating the analysis from this text on the identifiable models discussed Simpson and Maclaren [16], Borisov and Metelkin [22], and Boehm et al. [116] to assess the effectiveness of the approach under different conditions.

With the `IterativeBoundaryMethod`'s ability to have good coverage of the boundary of the profile, we find that only around 10-20 boundary points are required to capture all the uncertainty of the simultaneous bivariate profile when propagated into sets for predictive quantities.

7.4.3 Profile-Wise Prediction Sets From a Subset of Simultaneous Bivariate Profiles

We find that we can choose a smaller subset of simultaneous bivariate profiles that produce profile-wise prediction sets with a similar level of simultaneous coverage to the sets from the union of these profiles, with significantly lower computational cost. We demonstrate that a systematic selection has better coverage than picking randomly. However, for this to be more effective we need to choose profiles that are complementary, otherwise we could obtain results that are worse than the average random subset. We find these subsets by considering simulated pointwise coverage of profile-wise predictions sets from individual simultaneous bivariate profiles, which is computationally expensive, but effective.

Methods to systematically identify these small subsets in a lower cost fashion is an opportunity for future work. Methods could include: when considering the pointwise coverage of univariate profiles (instead of bivariate profiles), choosing parameter combinations that dominate the trajectory in successive time periods (if this occurs). This could also be considered using asymptotic analysis of model equations. If the trajectory has multiple model components that are weakly coupled then choosing bivariate parameter sets from the parameters in the equation of each component may be appropriate. Finally, PCA could be used to identify the parameters along the most stiff directions, which may impact the uncertainty in predictive quantities the most. However, this is a local approximation and may give inaccurate results if non-linearity is present.

7.4.4 Performance Against Full Parameter Confidence Sets

We find that when we use less conservative settings for the number of boundary points on the bivariate profile and the tolerance of the optimisation scheme, profile-wise prediction sets from the union of simultaneous bivariate profiles are always more computationally performant than sets from full parameter confidence set at the same level of simulated simultaneous coverage.

For the logistic model, we observed a 50% decrease in computational cost for the same level of coverage (the desired 0.95), with a further 30% decrease when only considering two simultaneous bivariate profiles.

For the Lotka-Volterra model, the sets from the union of simultaneous bivariate profiles do not reach the desired level of simultaneous coverage (0.85 [0.82, 0.87]), demonstrating the approximate nature of the profile-wise prediction sets. However, this is *better than* the 0.79 [0.77, 0.82] simultaneous coverage from the full parameter confidence set with the same computational cost and slightly lower than the 0.87 [0.85, 0.89] mustered for twice the cost. Additionally, when considering only four of the six simultaneous bivariate profiles, the computational cost of these bivariate profiles decreases by 32% for only a slight drop in simultaneous coverage (0.84, [0.82, 0.86]).

For the two-species logistic model, the profile-wise prediction sets from the union of simultaneous bivariate profiles slightly outperform or have the same point estimates of coverage as the sets from the full parameter confidence set. This is the case for both predictive quantities, at a 93% reduction in computational cost (0.92 [0.90, 0.93] vs 0.90 [0.84, 0.93] and 0.92 [0.90, 0.94] vs 0.92 [0.87, 0.95]). If the selected subset of four and five parameters were used for the respective predictive quantities, then only a tiny 1.3%-1.6% of the ten million likelihood calls would be required for effectively the same level of coverage. These computational outcomes would be even more stark if the less well-informed parameter bounds used for the profiles were also used for the sampled set.

7.4.5 Prediction Goals

In summary, when making predictions using the PWA workflow (at least for identifiable models), if your goal is to:

1. Predict the model trajectory curvewise, use the union of simultaneous bivariate profiles to form profile-wise trajectory confidence sets. These same sets formed from a subset of these simultaneous bivariate profiles likely have coverage close to or equal to those from the union of all these profiles.
2. Predict the $1 - \delta$ population reference set curvewise, use the union of simultaneous bivariate

profiles to form profile-wise $(1 - \delta, 1 - \alpha)$ reference tolerance sets. These same sets formed from a subset of these simultaneous bivariate profiles likely have coverage close to or equal to those from the union of all these profiles. If the data distribution standard deviation is unknown, then profiles that include this parameter will be the most important for predicting the $1 - \delta$ population reference set.

3. Predict observations simultaneously, we recommend reconsidering the feasibility of this goal.
4. Predict observations simultaneously with our adjusted goal, use the union of simultaneous univariate profiles or a subset of simultaneous bivariate profiles to form profile-wise $(1 - \delta, 1 - \alpha)$ reference tolerance sets.

7.5 Stochastic Models

This work has focused on models of the form ‘deterministic mathematical model + error model’ within the PWA workflow. However, this comes with assumptions that may be unrealistic or unable to represent inherently stochastic biological processes [13, 14, 120]. Therefore, providing an effective framework for using stochastic models within the PWA workflow is a valuable contribution. The primary difficulty of their inclusion is the high computational cost to evaluate an exact likelihood function (if even possible) or simply the high computational cost relative to approximate mechanistic descriptions of the same processes [13, 14, 113, 119] (e.g. SIR models [120]). A simple procedure to reduce this cost, which obtains an approximate likelihood function, is to use multivariate multiple linear regression alongside a mechanistic continuum limit model of the stochastic model to produce a statistical ‘meta-model’ or surrogate as in Simpson et al. [14]. An alternative is using a ‘generalised likelihood function’ as in Warne et al. [113].

The approach of Simpson et al. [14] is effectively the ‘Bayesian approximation error’ (BAE) approach [121–123], where a surrogate is used as a correction term to model the difference between the fine stochastic model and coarse mechanistic model. In this approach, we choose a set of design points in parameter space from a design distribution to simulate the coarse and fine models. The difference between the two models at these design points is then evaluated to form the surrogate model, which produces the log-likelihood function. Parameter inference and profile-wise analysis can then be performed using the mechanistic model paired with the surrogate, greatly reducing the computational cost.

A simple example of this would be the birth-death process model for population growth [124] as found in the *Experiments* folder of `UoA_MastersWorking`. This model has two parameters: the birth rate β and death rate δ . The total rate of each of these is βN and δN , where N is the current population size.

If we only observe the total population size over time, then when using this procedure, we see that both parameters are structurally non-identifiable (Figure 49a), as expected [43, 51]. Similarly, if we also observe the number of deaths over time, both parameters become practically identifiable (Figure 49b).

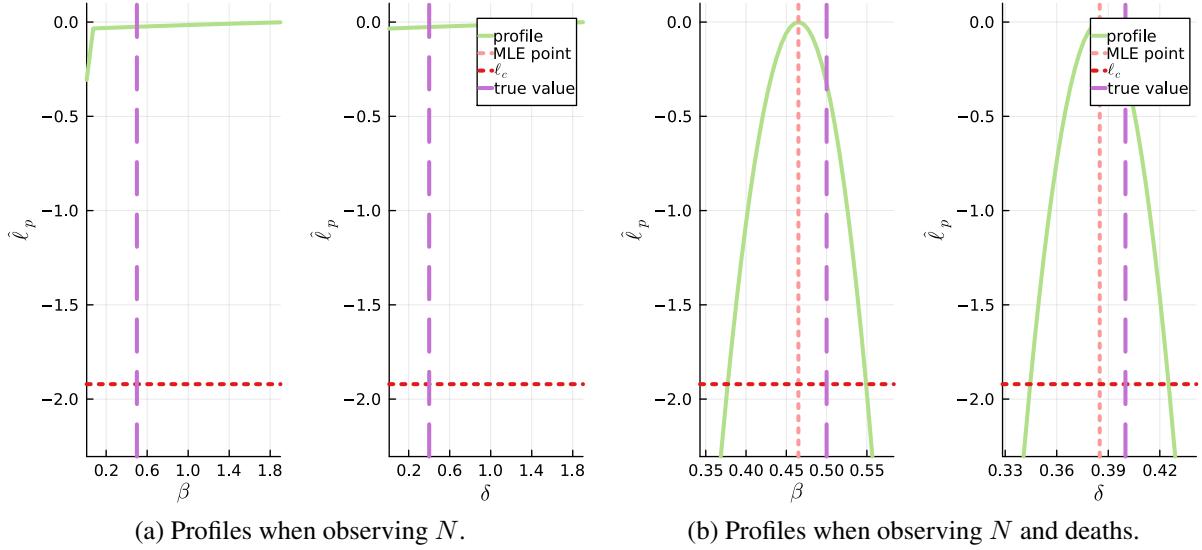


Figure 49: Normalised profile log-likelihood function (solid green) and ellipse approximation of that function (dashed light blue) for each parameter in a stochastic birth-death model when observing only N or N and the number of deaths. Parameter settings are $\alpha = 0.5$, $\beta = 0.4$, $N(t = 0) = 1000$ with observation time points $t_{1:I} = [0.1, 0.1 + (2.9 \div 99), \dots, 3.0]$. The vertical line (dotted light red) is the MLE value of each parameter. The horizontal line (dotted dark red) is the value of ℓ_c that corresponds to 95% likelihood-based confidence intervals for each parameter (intersection with the green line).

The specification of the design distribution used to form the surrogate impacts the degree of uncertainty observed for parameters and resultantly profile-wise predictions, which is a feature of this approach [14]. Additionally, we note that the simultaneous $(1 - \delta, 1 - \alpha)$ reference tolerance set we specify for our mechanistic models has a more murky definition when used with stochastic models. Namely, in a stochastic model, each observation depends on the previous observation. In contrast, in the ‘deterministic mathematical model + error model’, only the model trajectory component of the observation is dependent, while the error associated with each observation is independent. Resultantly, a curvewise population reference set in a stochastic model should potentially contain $1 - \delta$ of all possible stochastic trajectories with confidence $1 - \alpha$, which has the difficulties of simultaneous prediction sets (Section 2.3.6.2). Instead, it may be more appropriate to consider these as simultaneous population reference intervals, which expect to contain the $1 - \delta$ reference interval of stochastic observations at each time point simultaneously, with confidence $1 - \alpha$. However, this may introduce issues with pointwise coverage [105]. Furthermore, the design distribution and surrogate chosen may impact the reference

tolerance set's coverage performance. This motivates future work to consider the quality of surrogates used within the PWA workflow and evaluate reference tolerance intervals with these surrogates.

7.6 Research Objectives

We have successfully met the first five research objectives specified in Section 1.1 while discussing improvements, limitations and future work. Here, we summarise how these objectives have been met. We have provided a review of the PWA workflow and its existing applications in Section 2 while introducing and justifying the use of a different approach to quantifying uncertainty in observations using $(1 - \delta, 1 - \alpha)$ reference tolerance sets. We have provided a review of these tolerance intervals within the PWA workflow in Sections 2.2.8 and 2.3.6.2. The methods used to form univariate and bivariate profiles have been improved on those presented in Simpson and Maclaren [1], with additional improvements identified and suggested (Sections 3 and 4). To make the PWA workflow more accessible, these methods, alongside rejection sampling methods, profile-wise prediction sets, plotting capabilities and functions for easily testing the statistical performance of the approach, have been implemented in an open-source *Julia* package, `LikelihoodBasedProfileWiseAnalysis` [3]. We showed that, on identifiable models with low numbers of parameters, the asymptotic threshold used to form profiles should use $\nu = |\theta|$ to significantly improve the statistical performance of profile-wise predictions, which we refer to as profile paths for profile-wise analysis. Then, using the profile paths approach, we quantitatively demonstrated the value of the PWA workflow over rejection sampling approaches on existing examples in literature.

Comparing the PWA workflow's performance to a wider range of methods for uncertainty quantification of predictions remains valuable future work, for example, as in Villaverde et al. [45]. The value of using profiles to assess parameter identifiability has already been demonstrated [52].

7.7 Conclusions

We have qualitatively demonstrated the value of the PWA workflow while suggesting a modification to improve its statistical accuracy when making predictions on well-identified models at no additional computational cost. The approximate profile-wise sets for predictive quantities formed using the union of simultaneous bivariate profiles have similar levels of statistical accuracy to the prediction sets from full parameter confidence sets at up to a 93% decrease in computational cost on the models considered. This can be further decreased by considering a good subset of these simultaneous bivariate profiles. Moreover, these sets for predictive quantities have approximate simultaneous guarantees rather than only the pointwise guarantees of other efficient methods in a likelihood-based context [10, 11]. Additionally, when using the simultaneous bivariate profiles, relatively well-informed parameter

bounds are not required to observe reasonable performance levels. Finally, we observe that even the approximate profile-wise (95%, 95%) reference tolerance set from the union of simultaneous univariate profiles can be appropriate for capturing at least 95% of observations 95% of the time. If this goal is desired, then this provides an efficient method for quantifying uncertainty in observations.

We recommend repeating the analysis of the profile path approach on non-identifiable models and higher dimensional models, such as those in Simpson and Maclaren [16], Borisov and Metelkin [22], and Boehm et al. [116], to assess the effectiveness of the approach under different conditions. Additionally, we recommend further investigation into methods for systematically and efficiently identifying subsets of (bivariate) profiles with similar levels of statistical accuracy to the union of all profiles.

Moreover, we have implemented the PWA workflow as an open-source *Julia* package [3]. This makes using this workflow on additional models much more accessible while providing a systematic approach for evaluating expected performance. The methods for evaluating univariate and bivariate profiles, as implemented in `LikelihoodBasedProfileWiseAnalysis` [3], have been improved from those in Simpson and Maclaren [1]. Additionally, they are efficient enough to demonstrate the computational advantage of this workflow conclusively. Nonetheless, performance improvements can be made, which is valuable future work. In particular, using `LikelihoodProfiler`'s CICO method for both one-dimensional profiles and an extension into one-dimensional lines in bivariate profile space could greatly improve computational performance, especially for higher-dimensional models. Alternatively, implementing a bespoke bracketing method to reduce duplicate optimisation calls represents a simpler solution, albeit without improving performance to the same degree. Implementing procedures for calibrating the asymptotic threshold as used for full likelihood methods is also of interest.

Using the PWA workflow with stochastic models remains an important consideration for future work, particularly regarding the goal of predicting realisations. We provide an initial discussion of reference tolerance intervals and sets when using stochastic models, which can be built upon in future work.

References

- [1] Matthew J. Simpson and Oliver J. Maclarens. “Profile-Wise Analysis: A profile likelihood-based workflow for identifiability analysis, estimation, and prediction with mechanistic mathematical models”. In: *PLOS Computational Biology* 19.9 (Sept. 2023), pp. 1–31. doi: [10.1371/journal.pcbi.1011515](https://doi.org/10.1371/journal.pcbi.1011515).
- [2] Matthew J. Simpson et al. “Profile likelihood-based parameter and predictive interval analysis guides model choice for ecological population dynamics”. In: *Mathematical Biosciences* 355 (Jan. 1, 2023), p. 108950. ISSN: 0025-5564. doi: [10.1016/j.mbs.2022.108950](https://doi.org/10.1016/j.mbs.2022.108950).
- [3] Joel Trent. *LikelihoodBasedProfileWiseAnalysis.jl*. Version 1.0.0–DEV. Feb. 2024. URL: <https://github.com/JoelTrent/LikelihoodBasedProfileWiseAnalysis.jl>.
- [4] Joel Trent. *EllipseSampling.jl*. Version 0.2.0. Jan. 2024. URL: <https://github.com/JoelTrent/EllipseSampling.jl>.
- [5] Joel Trent. *UnivariateUnimodalHighestDensityRegion.jl*. Version 0.1.0. Nov. 2023. URL: <https://github.com/JoelTrent/UnivariateUnimodalHighestDensityRegion.jl>.
- [6] Léo Grinsztajn et al. “Bayesian workflow for disease transmission modeling in Stan”. In: *Statistics in Medicine* 40.27 (2021). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.9164>, pp. 6209–6234. ISSN: 1097-0258. doi: [10.1002/sim.9164](https://doi.org/10.1002/sim.9164).
- [7] Andreas Raue et al. “Joining forces of Bayesian and frequentist methodology: a study for inference in the presence of non-identifiability”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984 (Feb. 13, 2013). Publisher: Royal Society, p. 20110544. doi: [10.1098/rsta.2011.0544](https://doi.org/10.1098/rsta.2011.0544).
- [8] Luiz Hespanhol et al. “Understanding and interpreting confidence and credible intervals around effect estimates”. In: *Brazilian Journal of Physical Therapy* 23.4 (2019), pp. 290–301. ISSN: 1413-3555. doi: [10.1016/j.bjpt.2018.12.006](https://doi.org/10.1016/j.bjpt.2018.12.006).
- [9] Ryan J. Murphy, Oliver J. Maclarens, and Matthew J. Simpson. *Implementing measurement error models in a likelihood-based framework for estimation, identifiability analysis, and prediction in the life sciences*. July 4, 2023. doi: [10.48550/arXiv.2307.01539](https://arxiv.org/abs/2307.01539). arXiv: 2307.01539[q-bio, stat].
- [10] Clemens Kreutz, Andreas Raue, and Jens Timmer. “Likelihood based observability analysis and confidence intervals for predictions of dynamic models”. In: *BMC Systems Biology* 6.1 (Sept. 5, 2012), p. 120. ISSN: 1752-0509. doi: [10.1186/1752-0509-6-120](https://doi.org/10.1186/1752-0509-6-120).
- [11] Helge Hass et al. “Fast integration-based prediction bands for ordinary differential equation models”. In: *Bioinformatics* 32.8 (Apr. 15, 2016), pp. 1204–1210. ISSN: 1367-4803. doi: [10.1093/bioinformatics/btv743](https://doi.org/10.1093/bioinformatics/btv743).
- [12] Matthew J. Simpson et al. “Practical parameter identifiability for spatio-temporal models of cell invasion”. In: *Journal of The Royal Society Interface* 17.164 (Mar. 4, 2020). Publisher: Royal Society. doi: [10.1098/rsif.2020.0055](https://doi.org/10.1098/rsif.2020.0055).
- [13] Matthew J. Simpson et al. “Profile likelihood analysis for a stochastic model of diffusion in heterogeneous media”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 477.2250 (June 9, 2021). Publisher: Royal Society, p. 20210214. doi: [10.1098/rspa.2021.0214](https://doi.org/10.1098/rspa.2021.0214).
- [14] Matthew J. Simpson et al. “Reliable and efficient parameter estimation using approximate continuum limit descriptions of stochastic models”. In: *Journal of Theoretical Biology* 549 (Sept. 21, 2022), p. 111201. ISSN: 0022-5193. doi: [10.1016/j.jtbi.2022.111201](https://doi.org/10.1016/j.jtbi.2022.111201).
- [15] Ryan J. Murphy et al. “Computationally efficient framework for diagnosing, understanding and predicting biphasic population growth”. In: *Journal of The Royal Society Interface* 19.197 (Dec. 7, 2022). Publisher: Royal Society, p. 20220560. doi: [10.1098/rsif.2022.0560](https://doi.org/10.1098/rsif.2022.0560).
- [16] Matthew J. Simpson and Oliver J. Maclarens. *Making predictions using poorly identified mathematical models*. Nov. 27, 2023. doi: [10.1101/2023.11.26.568740](https://doi.org/10.1101/2023.11.26.568740). URL: <https://www.biorxiv.org/content/10.1101/2023.11.26.568740v1>.

- [17] Matthew J. Simpson, Ryan J. Murphy, and Oliver J. Maclaren. *Modelling count data with partial differential equation models in biology*. Sept. 13, 2023. doi: [10.1101/2023.09.09.556963](https://doi.org/10.1101/2023.09.09.556963).
- [18] E. M. Wright and P. Royston. “Calculating reference intervals for laboratory measurements”. In: *Statistical Methods in Medical Research* 8.2 (June 1999), pp. 93–112. issn: 0962-2802. doi: [10.1177/096228029900800202](https://doi.org/10.1177/096228029900800202).
- [19] Hormuzd A. Katki, Eric A. Engels, and Philip S. Rosenberg. “Assessing uncertainty in reference intervals via tolerance intervals: application to a mixed model describing HIV infection”. In: *Statistics in Medicine* 20 (Oct. 30, 2005), pp. 3185–3198. issn: 0277-6715. doi: [10.1002/sim.2171](https://doi.org/10.1002/sim.2171).
- [20] Daniel John VandenHeuvel. *ProfileLikelihood.jl*. Version 0.3.1. Language: eng. June 15, 2023. doi: [10.5281/zenodo.8042729](https://doi.org/10.5281/zenodo.8042729). url: <https://zenodo.org/record/8042729>.
- [21] Jeff Bezanson et al. “Julia: A Fresh Approach to Numerical Computing”. In: *SIAM Review* 59.1 (2017). Publisher: Society for Industrial and Applied Mathematics, pp. 65–98. issn: 0036-1445. doi: [10.1137/141000671](https://doi.org/10.1137/141000671).
- [22] Ivan Borisov and Evgeny Metelkin. “Confidence intervals by constrained optimization—An algorithm and software package for practical identifiability analysis in systems biology”. In: *PLOS Computational Biology* 16.12 (Dec. 21, 2020). Ed. by Daniel A. Beard, e1008495. issn: 1553-7358. doi: [10.1371/journal.pcbi.1008495](https://doi.org/10.1371/journal.pcbi.1008495).
- [23] Rafael Arutjunjan. *InformationGeometry.jl*. Version 1.14.4. Apr. 11, 2023. doi: [10.5281/zenodo.7817620](https://doi.org/10.5281/zenodo.7817620).
- [24] Yudi Pawitan. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford University Press, 2001. 528 pp. isbn: 978-0-19-850765-9.
- [25] Stephen B. Vardeman. “What about the Other Intervals?” In: *The American Statistician* 46.3 (1992), pp. 193–197. issn: 0003-1305. doi: [10.2307/2685212](https://doi.org/10.2307/2685212).
- [26] Naomi Altman and Martin Krzywinski. “Predicting with confidence and tolerance”. In: *Nature Methods* 15.11 (Nov. 1, 2018). Number: 11 Publisher: Nature Publishing Group, pp. 843–845. issn: 1548-7105. doi: [10.1038/s41592-018-0196-7](https://doi.org/10.1038/s41592-018-0196-7).
- [27] D. A. Sprott. *Statistical Inference in Science*. Springer Science & Business Media, Jan. 28, 2008. 254 pp. isbn: 978-0-387-22766-5.
- [28] Dahua Lin et al. *JuliaStats/Distributions.jl: v0.25.100*. Aug. 2023. doi: [10.5281/zenodo.8224988](https://doi.org/10.5281/zenodo.8224988).
- [29] A. Raue et al. “Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood”. In: *Bioinformatics* 25.15 (Aug. 1, 2009), pp. 1923–1929. issn: 1367-4803. doi: [10.1093/bioinformatics/btp358](https://doi.org/10.1093/bioinformatics/btp358).
- [30] Rob J. Hyndman. “Computing and Graphing Highest Density Regions”. In: *The American Statistician* 50.2 (May 1996). Publisher: [American Statistical Association, Taylor & Francis, Ltd.], pp. 120–126. issn: 0003-1305. doi: [10.2307/2684423](https://doi.org/10.2307/2684423).
- [31] Rupert G. Miller. *Simultaneous Statistical Inference*. Springer Series in Statistics. New York, NY: Springer, 1981. isbn: 978-1-4613-8124-2. doi: [10.1007/978-1-4613-8122-8](https://doi.org/10.1007/978-1-4613-8122-8). url: <http://link.springer.com/10.1007/978-1-4613-8122-8>.
- [32] Glen A. Satten. “Upper and Lower Bound Distributions that Give Simultaneous Confidence Intervals for Quantiles”. In: *Journal of the American Statistical Association* 90.430 (1995). Publisher: [American Statistical Association, Taylor & Francis, Ltd.], pp. 747–752. issn: 0162-1459. doi: [10.2307/2291087](https://doi.org/10.2307/2291087). url: <https://www.jstor.org/stable/2291087>.
- [33] Vaibhav Kumar Dixit and Christopher Rackauckas. *Optimization.jl: A Unified Optimization Package*. Version 3.12.1. Mar. 15, 2023. doi: [10.5281/zenodo.7738525](https://doi.org/10.5281/zenodo.7738525).
- [34] Steven G. Johnson. *The {NLopt} nonlinear-optimization package*. 2007. url: <https://github.com/stevengj/nlopt>.
- [35] Michael JD Powell. “The BOBYQA algorithm for bound constrained optimization without derivatives”. In: *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge* 26 (2009).

- [36] Keegan E. Hines, Thomas R. Middendorf, and Richard W. Aldrich. “Determination of parameter identifiability in nonlinear biophysical models: A Bayesian approach”. In: *Journal of General Physiology* 143.3 (Feb. 10, 2014), pp. 401–416. issn: 0022-1295. doi: [10.1085/jgp.20131116](https://doi.org/10.1085/jgp.20131116).
- [37] Elena Sellentin, Miguel Quartin, and Luca Amendola. “Breaking the spell of Gaussianity: forecasting with higher order Fisher matrices”. In: *Monthly Notices of the Royal Astronomical Society* 441.2 (June 21, 2014), pp. 1831–1840. issn: 1365-2966, 0035-8711. doi: [10.1093/mnras/stu689](https://doi.org/10.1093/mnras/stu689).
- [38] *Nonlinear Regression Analysis and Its Applications*. In collab. with Douglas M. Bates and Donald G. Watts. Series Title: Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., Aug. 26, 1988. isbn: 978-0-470-31675-7. doi: [10.1002/9780470316757.ch2](https://doi.org/10.1002/9780470316757.ch2).
- [39] Julia Diff. *JuliaDiff - Differentiation tools in Julia*. URL: <https://juliadiff.org/>.
- [40] Automatic differentiation. In: *Wikipedia*. Page Version ID: 1170351857. URL: https://en.wikipedia.org/w/index.php?title=Automatic_differentiation&oldid=1170351857.
- [41] Jarrett Revels, Miles Lubin, and Theodore Papamarkou. *Forward-Mode Automatic Differentiation in Julia*. July 26, 2016. arXiv: 1607.07892[cs]. URL: <https://arxiv.org/abs/1607.07892>.
- [42] Michael Friendly, Georges Monette, and John Fox. “Elliptical Insights: Understanding Statistical Methods through Elliptical Geometry”. In: *Statistical Science* 28.1 (2013). Publisher: Institute of Mathematical Statistics, pp. 1–39. issn: 0883-4237. URL: <https://www.jstor.org/stable/43288410>.
- [43] Diana Cole. *Parameter Redundancy and Identifiability*. New York: Chapman and Hall/CRC, May 10, 2020. 272 pp. isbn: 978-1-315-12000-3. doi: [10.1201/9781315120003](https://doi.org/10.1201/9781315120003).
- [44] Ali Shahmohammadi and Kimberley B. McAuley. “Sequential Model-Based A-Optimal Design of Experiments When the Fisher Information Matrix Is Noninvertible”. In: *Industrial & Engineering Chemistry Research* 58.3 (2019), pp. 1244–1261. doi: [10.1021/acs.iecr.8b03047](https://doi.org/10.1021/acs.iecr.8b03047).
- [45] Alejandro F. Villaverde et al. “Assessment of Prediction Uncertainty Quantification Methods in Systems Biology”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 20.3 (2023), pp. 1725–1736. issn: 1557-9964. doi: [10.1109/TCBB.2022.3213914](https://doi.org/10.1109/TCBB.2022.3213914).
- [46] Elena Sellentin. *DALI*. original-date: 2015-04-28T12:38:43Z. June 17, 2015. URL: <https://github.com/Lnasellentin/DALI>.
- [47] Elena Sellentin. “A fast, always positive definite and normalizable approximation of non-Gaussian likelihoods”. In: *Monthly Notices of the Royal Astronomical Society* 453.1 (Oct. 11, 2015), pp. 893–898. issn: 0035-8711, 1365-2966. doi: [10.1093/mnras/stv1671](https://doi.org/10.1093/mnras/stv1671). arXiv: 1506.04866[astro-ph, physics:hep-ex, physics:physics].
- [48] A. Raue et al. “Identifiability and observability analysis for experimental design in nonlinear dynamical models”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 20.4 (Dec. 30, 2010), p. 045105. issn: 1054-1500. doi: [10.1063/1.3528102](https://doi.org/10.1063/1.3528102).
- [49] Xabier Rey Barreiro and Alejandro F Villaverde. “Benchmarking tools for a priori identifiability analysis”. In: *Bioinformatics* 39.2 (Feb. 1, 2023), btad065. issn: 1367-4811. doi: [10.1093/bioinformatics/btad065](https://doi.org/10.1093/bioinformatics/btad065).
- [50] Oana-Teodora Chis et al. “On the relationship between sloppiness and identifiability”. In: *Mathematical Biosciences* 282 (Dec. 1, 2016), pp. 147–161. issn: 0025-5564. doi: [10.1016/j.mbs.2016.10.009](https://doi.org/10.1016/j.mbs.2016.10.009).
- [51] Alexander P. Browning et al. “Identifiability analysis for stochastic differential equation models in systems biology”. In: *Journal of the Royal Society, Interface* 17.173 (Dec. 2020), p. 20200652. issn: 1742-5662. doi: [10.1098/rsif.2020.0652](https://doi.org/10.1098/rsif.2020.0652).
- [52] Fabian Fröhlich, Fabian J. Theis, and Jan Hasenauer. “Uncertainty Analysis for Non-identifiable Dynamical Systems: Profile Likelihoods, Bootstrapping and More”. In: *Computational Methods in Systems Biology*. Ed. by Pedro Mendes, Joseph O. Dada, and Kieran Smallbone. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 61–72. isbn: 978-3-319-12982-2. doi: [10.1007/978-3-319-12982-2_5](https://doi.org/10.1007/978-3-319-12982-2_5).

- [53] Ruiwen Dong et al. “Differential Elimination for Dynamical Models via Projections with Applications to Structural Identifiability”. In: *SIAM Journal on Applied Algebra and Geometry* 7.1 (Mar. 31, 2023). Publisher: Society for Industrial and Applied Mathematics, pp. 194–235. doi: [10.1137/22M1469067](https://doi.org/10.1137/22M1469067).
- [54] Hoon Hong et al. “SIAN: software for structural identifiability analysis of ODE models”. In: *Bioinformatics* 35.16 (Aug. 15, 2019), pp. 2873–2874. issn: 1367-4803. doi: [10.1093/bioinformatics/bty1069](https://doi.org/10.1093/bioinformatics/bty1069).
- [55] Hoon Hong et al. “Global Identifiability of Differential Models”. In: *Communications on Pure and Applied Mathematics* 73.9 (Sept. 2020). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.21921>, pp. 1831–1879. issn: 1097-0312. doi: [10.1002/cpa.21921](https://doi.org/10.1002/cpa.21921).
- [56] Sandra Díaz-Seoane, Xabier Rey Barreiro, and Alejandro F Villaverde. “STRIKE-GOLDD 4.0: user-friendly, efficient analysis of structural identifiability and observability”. In: *Bioinformatics* 39.1 (Jan. 1, 2023), btac748. issn: 1367-4811. doi: [10.1093/bioinformatics/btac748](https://doi.org/10.1093/bioinformatics/btac748).
- [57] Alexey Ovchinnikov et al. “Computing all identifiable functions of parameters for ODE models”. In: *Systems & Control Letters* 157 (Nov. 1, 2021), p. 105030. issn: 0167-6911. doi: [10.1016/j.sysconle.2021.105030](https://doi.org/10.1016/j.sysconle.2021.105030).
- [58] Matthew J. Simpson et al. “Parameter identifiability and model selection for sigmoid population growth models”. In: *Journal of Theoretical Biology* 535 (Feb. 21, 2022), p. 110998. issn: 0022-5193. doi: [10.1016/j.jtbi.2021.110998](https://doi.org/10.1016/j.jtbi.2021.110998).
- [59] A. Raue et al. “Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems”. In: *Bioinformatics* 31.21 (Nov. 1, 2015), pp. 3558–3560. issn: 1367-4803. doi: [10.1093/bioinformatics/btv405](https://doi.org/10.1093/bioinformatics/btv405).
- [60] Clemens Kreutz. “An easy and efficient approach for testing identifiability”. In: *Bioinformatics* 34.11 (June 1, 2018), pp. 1913–1921. issn: 1367-4803. doi: [10.1093/bioinformatics/bty035](https://doi.org/10.1093/bioinformatics/bty035).
- [61] Amin Ghadami and Bogdan I. Epureanu. “Data-driven prediction in dynamical systems: recent developments”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 380.2229 (June 20, 2022). Publisher: Royal Society, p. 20210213. doi: [10.1098/rsta.2021.0213](https://doi.org/10.1098/rsta.2021.0213).
- [62] Thomas J. Snowden, Piet H. van der Graaf, and Marcus J. Tindall. “Methods of Model Reduction for Large-Scale Biological Systems: A Survey of Current Methods and Trends”. In: *Bulletin of Mathematical Biology* 79.7 (July 1, 2017), pp. 1449–1486. issn: 1522-9602. doi: [10.1007/s11538-017-0277-2](https://doi.org/10.1007/s11538-017-0277-2).
- [63] M. Joshi, A. Seidel-Morgenstern, and A. Kremling. “Exploiting the bootstrap method for quantifying parameter confidence intervals in dynamical systems”. In: *Metabolic Engineering* 8.5 (Sept. 1, 2006), pp. 447–455. issn: 1096-7176. doi: [10.1016/j.ymben.2006.04.003](https://doi.org/10.1016/j.ymben.2006.04.003).
- [64] Clemens Kreutz et al. “Profile likelihood in systems biology”. In: *The FEBS Journal* 280.11 (June 2013), pp. 2564–2571. issn: 1742464X. doi: [10.1111/febs.12276](https://doi.org/10.1111/febs.12276).
- [65] Dean S. Oliver. “Metropolized Randomized Maximum Likelihood for Improved Sampling from Multimodal Distributions”. In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017). Publisher: Society for Industrial and Applied Mathematics, pp. 259–277. doi: [10.1137/15M1033320](https://doi.org/10.1137/15M1033320).
- [66] D. J. Venzon and S. H. Moolgavkar. “A Method for Computing Profile-Likelihood-Based Confidence Intervals”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 37.1 (1988). Publisher: [Wiley, Royal Statistical Society], pp. 87–94. issn: 0035-9254. doi: [10.2307/2347496](https://doi.org/10.2307/2347496).
- [67] Jian-Shen Chen and Robert I. Jennrich. “Simple Accurate Approximation of Likelihood Profiles”. In: *Journal of Computational and Graphical Statistics* 11.3 (Sept. 2002). Publisher: [American Statistical Association, Taylor & Francis, Ltd., Institute of Mathematical Statistics, Interface Foundation of America], pp. 714–732. issn: 1061-8600. doi: [10.1198/106186002493](https://doi.org/10.1198/106186002493).

- [68] R. Boiger et al. “Integration based profile likelihood calculation for PDE constrained parameter estimation problems”. In: *Inverse Problems* 32.12 (Nov. 18, 2016). Publisher: IOP Publishing, p. 125009. ISSN: 0266-5611. doi: [10.1088/0266-5611/32/12/125009](https://doi.org/10.1088/0266-5611/32/12/125009).
- [69] Paul Stapor, Fabian Fröhlich, and Jan Hasenauer. “Optimization and profile calculation of ODE models using second order adjoint sensitivity analysis”. In: *Bioinformatics* 34.13 (July 2018), pp. i151–i159. ISSN: 1367-4803. doi: [10.1093/bioinformatics/bty230](https://doi.org/10.1093/bioinformatics/bty230).
- [70] Paul Stapor et al. “PESTO: Parameter EStimation TOolbox”. In: *Bioinformatics* 34.4 (Feb. 15, 2018), pp. 705–707. ISSN: 1367-4803. doi: [10.1093/bioinformatics/btx676](https://doi.org/10.1093/bioinformatics/btx676).
- [71] Daniel Kaschek et al. “Dynamic Modeling, Parameter Estimation, and Uncertainty Analysis in R”. In: *Journal of Statistical Software* 88.10 (Apr. 30, 2019), pp. 1–32. ISSN: 1548-7660. doi: [10.18637/jss.v088.i10](https://doi.org/10.18637/jss.v088.i10).
- [72] Andrew R. Conn, Nicholas I. M. Gould, and Philippe Toint. “A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds”. In: *SIAM Journal on Numerical Analysis* 28.2 (1991). Publisher: Society for Industrial and Applied Mathematics, pp. 545–572. doi: [10.1137/0728030](https://doi.org/10.1137/0728030).
- [73] E.G. Birgin and J.M. Martínez. “Improving ultimate convergence of an augmented Lagrangian method”. In: *Optimization Methods and Software* 23.2 (2008). Publisher: Taylor & Francis, pp. 177–195. doi: [10.1080/10556780701577730](https://doi.org/10.1080/10556780701577730).
- [74] François Bachoc, Tommaso R Cesari, and Sébastien Gerchinovitz. “The sample complexity of level set approximation”. In: *AISTATS 2021*. Vol. 130. Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. virtual, United States, Apr. 2021, pp. 424–432. URL: <https://hal.science/hal-02976018>.
- [75] Lukas Alexander Heinrich, Gilles Louppe, and Kyle Stuart Cranmer. “Excursion Set Estimation using Sequential Entropy Reduction for Efficient Searches for New Physics at the LHC”. In: *Indico*. 19th International Workshop on Advanced Computing and Analysis Techniques in Physics Research. Mar. 13, 2019. URL: <https://indico.cern.ch/event/708041/contributions/3269754/>.
- [76] Rafael Arutjunjan, Bjoern Malte Schaefer, and Clemens Kreutz. *Constructing Exact Confidence Regions on Parameter Manifolds of Non-Linear Models*. Nov. 7, 2022. doi: [10.48550/arXiv.2211.03421](https://doi.org/10.48550/arXiv.2211.03421). arXiv: [2211.03421\[physics, stat\]](https://arxiv.org/abs/2211.03421).
- [77] Alkis Gotovos et al. “Active learning for level set estimation”. In: *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. IJCAI ’13*. Beijing, China: AAAI Press, Aug. 3, 2013, pp. 1344–1350. ISBN: 978-1-57735-633-2. doi: [10.5555/2540128.2540322](https://doi.org/10.5555/2540128.2540322).
- [78] Andrea Zanette, Junzi Zhang, and Mykel J. Kochenderfer. “Robust Super-Level Set Estimation Using Gaussian Processes”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Michele Berlingario et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, Jan. 2019, pp. 276–291. ISBN: 978-3-030-10928-8. doi: [10.1007/978-3-030-10928-8_17](https://doi.org/10.1007/978-3-030-10928-8_17).
- [79] Dario Azzimonti et al. “Adaptive Design of Experiments for Conservative Estimation of Excursion Sets”. In: *Technometrics* 63.1 (2021), pp. 13–26. doi: [10.1080/00401706.2019.1693427](https://doi.org/10.1080/00401706.2019.1693427).
- [80] David Wu et al. “Likelihood-based estimation and prediction for a measles outbreak in Samoa”. In: *Infectious Disease Modelling* 8.1 (Mar. 2023), pp. 212–227. ISSN: 2468-0427. doi: [10.1016/j.idm.2023.01.007](https://doi.org/10.1016/j.idm.2023.01.007).
- [81] Tina Toni et al. “Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems”. In: *Journal of The Royal Society Interface* 6.31 (July 9, 2008), pp. 187–202. doi: [10.1098/rsif.2008.0172](https://doi.org/10.1098/rsif.2008.0172).
- [82] Mikael Sunnåker et al. “Approximate Bayesian Computation”. In: *PLOS Computational Biology* 9.1 (2013). Publisher: Public Library of Science, e1002803. ISSN: 1553-7358. doi: [10.1371/journal.pcbi.1002803](https://doi.org/10.1371/journal.pcbi.1002803).

- [83] Scott A. Sisson, Yanan Fan, and Mark Beaumont, eds. *Handbook of Approximate Bayesian Computation*. 1st ed. New York: Chapman and Hall/CRC, Sept. 15, 2018. 678 pp. ISBN: 978-1-315-11719-5. doi: [10.1201/9781315117195](https://doi.org/10.1201/9781315117195).
- [84] Ernesto Bonomi and Jean-Luc Lutton. “The N-City Travelling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm”. In: *SIAM Review* 26.4 (Oct. 1984). Publisher: Society for Industrial and Applied Mathematics, pp. 551–568. ISSN: 0036-1445. URL: <https://www.jstor.org/stable/2030978>.
- [85] Evan Fields. *TravelingSalesmanHeuristics.jl*. Version 0.3.3. May 29, 2020. URL: <https://github.com/evanfields/TravelingSalesmanHeuristics.jl/tree/master>.
- [86] Thomas H. Cormen et al. *Introduction to Algorithms*. 3rd ed. MIT Press, July 31, 2009. ISBN: 978-0-262-27083-0.
- [87] R. L. Graham. “An efficient algorithm for determining the convex hull of a finite planar set”. In: *Information Processing Letters* 1.4 (June 1, 1972), pp. 132–133. ISSN: 0020-0190. doi: [10.1016/0020-0190\(72\)90045-2](https://doi.org/10.1016/0020-0190(72)90045-2).
- [88] JuliaGeometry/Meshes.jl. Version 0.34.12. Aug. 2023. URL: <https://github.com/JuliaGeometry/Meshes.jl/tree/master>.
- [89] Adriano Moreira and Maribel Santos. “Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points.” In: Jan. 1, 2007, pp. 61–68.
- [90] Luke Stagner. *ConcaveHull.jl*. Version 1.2.0. Mar. 29, 2021. URL: <https://github.com/lstagner/ConcaveHull.jl>.
- [91] Fausto Bernardini and Chandrajit Bajaj. “Sampling and Reconstructing Manifolds Using Alpha-Shapes”. In: *Proceedings of the 9th Canadian Conference on Computational Geometry* (Aug. 1997), pp. 193–198. URL: <https://cccg.ca/proceedings/1997/>.
- [92] Tran Kai Frank Da. *2D Alpha Shapes*. 5.6. booktitle: {CGAL} User and Reference Manual. CGAL Editorial Board, 2023. URL: <https://doc.cgal.org/latest/Manual/packages.html#PkgAlphaShapes2>.
- [93] Alexandru Suciu. “MTH U565 Topology: Homotopy”. Lecture. Lecture. North Eastern University, 2008. URL: <https://web.northeastern.edu/suciu/U565/U565sp10-homotopy.pdf>.
- [94] Numerical algebraic geometry. In: *Wikipedia*. Page Version ID: 1170218249. Aug. 13, 2023. URL: https://en.wikipedia.org/w/index.php?title=Numerical_algebraic_geometry&oldid=1170218249#Homotopy_continuation.
- [95] Star domain. In: *Wikipedia*. Page Version ID: 1137885042. Feb. 2023. URL: https://en.wikipedia.org/w/index.php?title=Star_domain&oldid=1137885042.
- [96] Daniel Garcia-Castellanos and Umberto Lombardo. “Poles of inaccessibility: A calculation algorithm for the remotest places on earth”. In: *Scottish Geographical Journal* 123.3 (Sept. 1, 2007), pp. 227–233. ISSN: 1470-2541. doi: [10.1080/14702540801897809](https://doi.org/10.1080/14702540801897809).
- [97] Volodymyr Agafonkin. *Polylabel: a fast algorithm for finding the pole of inaccessibility of a polygon*. Version 1.2.0. July 11, 2016. URL: <https://github.com/mapbox/polylabel>.
- [98] Alejandro F. Villaverde et al. “A consensus approach for estimating the predictive accuracy of dynamic models in biology”. In: *Computer Methods and Programs in Biomedicine* 119.1 (Apr. 1, 2015), pp. 17–28. ISSN: 0169-2607. doi: [10.1016/j.cmpb.2015.02.001](https://doi.org/10.1016/j.cmpb.2015.02.001).
- [99] H. Sulieman, P. J. McLellan, and D. W. Bacon. “A Profile-Based Approach to Parametric Sensitivity Analysis of Nonlinear Regression Models”. In: *Technometrics* 43.4 (2001). Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], pp. 425–433. ISSN: 0040-1706. URL: <https://www.jstor.org/stable/1270813>.
- [100] H Sulieman, P. J McLellan, and D. W Bacon. “A profile-based approach to parametric sensitivity in multiresponse regression models”. In: *Computational Statistics & Data Analysis* 45.4 (May 10, 2004), pp. 721–740. ISSN: 0167-9473. doi: [10.1016/S0167-9473\(03\)00086-0](https://doi.org/10.1016/S0167-9473(03)00086-0).

- [101] Xiaoyu Dong and Thomas Mathew. “Central tolerance regions and reference regions for multivariate normal populations”. In: *Journal of Multivariate Analysis* 134 (Feb. 1, 2015), pp. 50–60. issn: 0047-259X. doi: 10.1016/j.jmva.2014.10.009.
- [102] Lianne Siegel et al. “A Guide to Estimating the Reference Range From a Meta-Analysis Using Aggregate or Individual Participant Data”. In: *American Journal of Epidemiology* 191.5 (Mar. 24, 2022), pp. 948–956. issn: 0002-9262. doi: 10.1093/aje/kwac013.
- [103] Wei Liu, Frank Bretz, and Mario Cortina-Borja. “Reference range: Which statistical intervals to use?” In: *Statistical Methods in Medical Research* 30.2 (Feb. 2021), pp. 523–534. issn: 0962-2802. doi: 10.1177/0962280220961793.
- [104] Steven De Gryze, Ivan Langhans, and Martina Vandebroek. “Using the correct intervals for prediction: A tutorial on tolerance intervals for ordinary least-squares regression”. In: *Chemometrics and Intelligent Laboratory Systems* 87.2 (June 15, 2007), pp. 147–154. issn: 0169-7439. doi: 10.1016/j.chemolab.2007.03.002. url: <https://www.sciencedirect.com/science/article/pii/S0169743907000457> (visited on 12/13/2023).
- [105] Jonas L. Juul et al. “Fixed-time descriptive statistics underestimate extremes of epidemic curve ensembles”. In: *Nature Physics* 17.1 (Jan. 2021). Number: 1 Publisher: Nature Publishing Group, pp. 5–8. issn: 1745-2481. doi: 10.1038/s41567-020-01121-y.
- [106] *JuliaStats/HypothesisTests.jl*. Version 0.11.0. June 1, 2023. url: <https://github.com/JuliaStats/HypothesisTests.jl>.
- [107] Jeremy Kepner et al. “Fast Mapping onto Census Blocks”. In: *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. 2020 IEEE High Performance Extreme Computing Conference (HPEC). ISSN: 2643-1971. Sept. 2020, pp. 1–8. doi: 10.1109/HPEC43674.2020.9286157.
- [108] Klaus Crusius and Rafael Schouten. *PolygonInbounds.jl*. Version 0.2.0. Apr. 26, 2020. url: <https://github.com/KlausC/PolygonInbounds.jl/tree/master> (visited on 08/29/2023).
- [109] John Verzani. *{Roots.jl}: Root finding functions for Julia*. 2020. url: <https://github.com/JuliaMath/Roots.jl>.
- [110] William H. Press et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. USA: Cambridge University Press, Sept. 10, 2007. isbn: 978-0-521-88068-8.
- [111] J. A. Nelder and A. Mead. “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. url: <https://doi.org/10.1093/comjnl/7.4.308>.
- [112] M. J. Box. “A New Method of Constrained Optimization and a Comparison With Other Methods”. In: *The Computer Journal* 8.1 (Apr. 1965), pp. 42–52. url: <https://doi.org/10.1093/comjnl/8.1.42>.
- [113] David J. Warne et al. “Generalised likelihood profiles for models with intractable likelihoods”. In: *Statistics and Computing* 34.1 (2024), p. 50. issn: 1573-1375. doi: 10.1007/s11222-023-10361-w.
- [114] Julie Bachmann et al. “Division of labor by dual feedback regulators controls JAK2/STAT5 signaling over broad ligand range”. In: *Molecular Systems Biology* 7.1 (Jan. 2011). Publisher: John Wiley & Sons, Ltd, p. 516. issn: 1744-4292. doi: 10.1038/msb.2011.50.
- [115] I. F. D. Oliveira and R. H. C. Takahashi. “An Enhancement of the Bisection Method Average Performance Preserving Minmax Optimality”. In: *ACM Transactions on Mathematical Software* 47.1 (Dec. 8, 2020), 5:1–5:24. issn: 0098-3500. doi: 10.1145/3423597.
- [116] Martin E. Boehm et al. “Identification of Isoform-Specific Dynamics in Phosphorylation-Dependent STAT5 Dimerization by Quantitative Mass Spectrometry and Mathematical Modeling”. In: *Journal of Proteome Research* 13.12 (Dec. 5, 2014). Publisher: American Chemical Society, pp. 5685–5694. issn: 1535-3893. doi: 10.1021/pr5006923.
- [117] *Logistic function, In ecology: modeling population growth*. In: *Wikipedia*. Page Version ID: 1188303275. Dec. 4, 2023. url: https://en.wikipedia.org/w/index.php?title=Logistic_function&oldid=1188303275#In_ecology:_modeling_population_growth.

- [118] Serafeim Loukas. *PCA clearly explained — How, when, why to use it and feature importance: A guide in Python*. Towards Data Science. May 31, 2020. url: <https://towardsdatascience.com/pca-clearly-explained-how-when-why-to-use-it-and-feature-importance-a-guide-in-python-7c274582c37e>.
- [119] David J. Warne, Ruth E. Baker, and Matthew J. Simpson. “Simulation and inference algorithms for stochastic biochemical reaction networks: from basic concepts to state-of-the-art”. In: *Journal of The Royal Society Interface* 16.151 (Feb. 27, 2019). Publisher: Royal Society, p. 20180943. doi: [10.1098/rsif.2018.0943](https://doi.org/10.1098/rsif.2018.0943).
- [120] Darren J. Wilkinson. *Stochastic Modelling for Systems Biology*. 1st ed. Chapman and Hall/CRC, Apr. 18, 2006. 280 pp. ISBN: 978-0-429-13800-3. doi: [10.1201/9781420010664](https://doi.org/10.1201/9781420010664).
- [121] Jari P. Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Red. by S.S. Antman, J.E. Marsden, and L. Sirovich. Vol. 160. Applied Mathematical Sciences. New York, NY: Springer, 2005. ISBN: 978-0-387-22073-4. doi: [10.1007/b138659](https://doi.org/10.1007/b138659).
- [122] Jari Kaipio and Erkki Somersalo. “Statistical inverse problems: Discretization, model reduction and inverse crimes”. In: *Journal of Computational and Applied Mathematics*. Special Issue: Applied Computational Inverse Problems 198.2 (Jan. 15, 2007), pp. 493–504. ISSN: 0377-0427. doi: [10.1016/j.cam.2005.09.027](https://doi.org/10.1016/j.cam.2005.09.027).
- [123] Jari Kaipio and Ville Kolehmainen. “32 Approximate marginalization over modelling errors and uncertainties in inverse problems”. In: *Bayesian Theory and Applications*. Ed. by Paul Damien et al. Oxford University Press, Jan. 24, 2013, pp. 644–672. ISBN: 978-0-19-969560-7. doi: [10.1093/acprof:oso/9780199695607.003.0032](https://doi.org/10.1093/acprof:oso/9780199695607.003.0032).
- [124] BingKan Xue. *Stochastic Processes: Birth-Death Process*. PHZ4710 - Introduction to Biological Physics. 2022. url: <https://cmp.phys.ufl.edu/PHZ4710/files/unit3/birth-death.html>.
- [125] Jeffrey Sarnoff. *AngleBetweenVectors.jl*. Version 0.3.1. June 25, 2022. url: <https://doi.org/10.5281/zenodo.6745689>.

Appendices

A Description of the Example Models

In this section, we provide the equations, parameter values and bounds used for each of the models considered in Sections 3, 4, 5 and 6. Most of these have been taken from other works [1, 2]. In particular, these are the logistic and Lotka-Volterra models with Gaussian data distributions from Simpson and Maclaren [1], the two-species logistic model of coral reefs with real data (site 1 data is used) from Simpson et al. [2], which also uses a Gaussian data distribution, and the Gaussian approximation of a binomial distribution. We also consider the case where the two-species logistic model uses a logit-normal data distribution.

A.1 Logistic Model

The logistic model with a Gaussian data distribution [1] has the following differential equation for the population density $C(t) \geq 0$:

$$\frac{dC(t)}{dt} = \lambda C(t) \left[1 - \frac{C(t)}{K} \right], \quad (33)$$

where the model parameter vector is given by $\theta^M = (\lambda, K, C(0))$. The corresponding additive Gaussian data distribution, with a fixed standard deviation, has a density function for the observed data given by:

$$y_i \sim p(y_i; \theta) \sim \mathcal{N}(z_i(\theta^M), \sigma_N^2), \quad (34)$$

where $z_i(\theta^M) = z(t_i; \theta^M)$ is the model solution of Equation 33 at t_i and $\sigma = 10$.

The true parameter values are $\theta^M = (0.01, 100, 10)$. The corresponding lower and upper parameter bounds are $a = (0, 50, 0)$ and $b = (0.05, 150, 50)$. Observation times are $t_{1:I} = 0, 100, 200, \dots, 1000$. Example observations, the true model trajectory and 95% population reference set under this parameterisation can be seen in Figure 50.

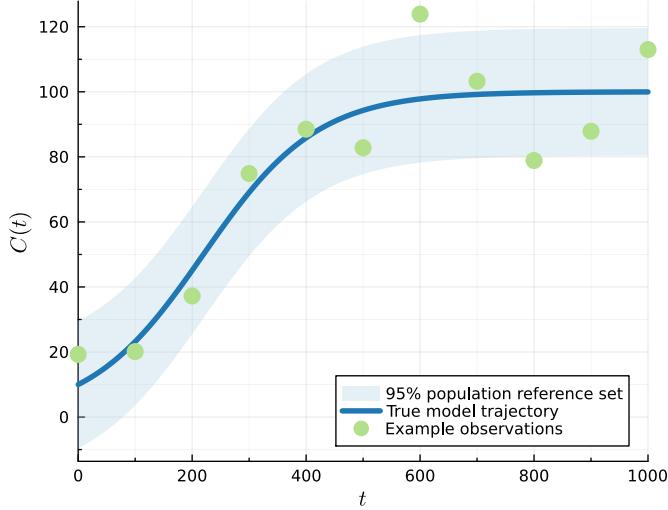


Figure 50: Logistic model trajectory, 95% population reference set and example observations [1].

A.2 Lotka-Volterra Model

The Lotka-Volterra model with a Gaussian data distribution [1] has the following differential equations for the population size of the prey species $x(t)$ and predator species $y(t)$:

$$\frac{dx(t)}{dt} = \alpha x(t) - x(t)y(t), \quad (35)$$

$$\frac{dy(t)}{dt} = \beta x(t)y(t) - y(t), \quad (36)$$

where the model parameter vector is given by $\theta^M = (\alpha, \beta, x(0), y(0))$. The corresponding additive Gaussian data distribution, with a fixed standard deviation, has a density function for the observed data given by:

$$y_i \sim p(y_i; \theta) \sim \mathcal{N}(z_i(\theta^M), \sigma_N^2 \mathbb{I}), \quad (37)$$

where $z_i(\theta^M) = z(t_i; \theta^M) = (x(t_i; \theta^M), y(t_i; \theta^M))$ from Equations 35 and 36, meaning at each t_i we have an observation of both $x(t)$ and $y(t)$, $y_i^o = (x_i^o, y_i^o)$, \mathbb{I} is a 2×2 identity matrix and $\sigma = 0.2$.

The true parameter values are $\theta^M = (0.9, 1.1, 0.8, 0.3)$. The corresponding lower and upper parameter bounds are $a = (0.7, 0.7, 0.5, 0.1)$ and $b = (1.2, 1.4, 1.2, 0.5)$. Observation times are $t_{1:I} = 0, 0.5, 1.0, \dots, 7$. The times considered for predictions are extended up to $t_I = 10$.

Example observations, the true model trajectory and 95% population reference set under this parameterisation can be seen in Figure 51. The time locations used for parameter estimation and prediction, and for predictions only, are shaded.

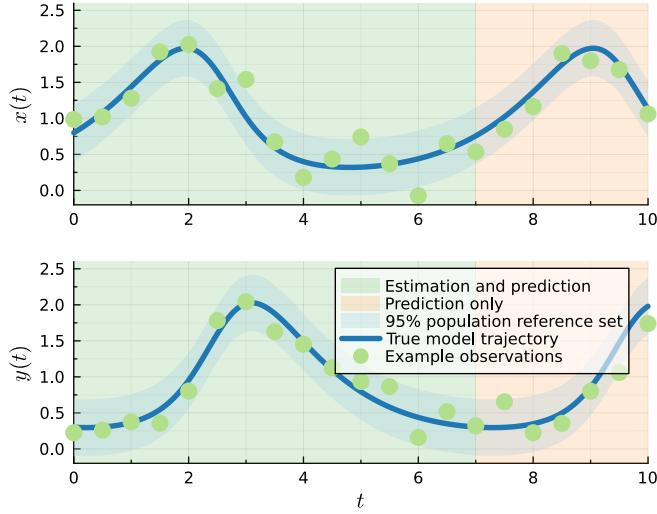


Figure 51: Lotka-Volterra model trajectory, 95% population reference set and example observations [1].

A.3 Two-Species Logistic Model

The two-species logistic model with a Gaussian data distribution [2] has the following differential equations for the population densities of the two species $C_1(t) \geq 0$ and $C_2(t) \geq 0$:

$$\frac{dC_1(t)}{dt} = \lambda_1 C_1(t) \left[1 - \frac{S(t)}{K} \right], \quad (38)$$

$$\frac{dC_2(t)}{dt} = \lambda_2 C_2(t) \left[1 - \frac{S(t)}{K} \right] - \delta C_2(t) \left[\frac{C_1(t)}{K} \right], \quad (39)$$

where $S(t) = C_1(t) + C_2(t)$, and the full parameter vector is given by $\theta = (\lambda_1, \lambda_2, K, \delta, C_1(0), C_2(0), \sigma)$. The corresponding additive Gaussian data distribution, with an estimated standard deviation, has a density function for the observed data given by:

$$y_i \sim p(y_i; \theta) \sim \mathcal{N}(z_i(\theta^M), \theta^o \mathbb{I}) \sim \mathcal{N}(z_i(\theta^M), \sigma_N^2 \mathbb{I}), \quad (40)$$

where $\theta^M = (\lambda_1, \lambda_2, K, \delta, C_1(0), C_2(0))$, $\theta^o = \sigma$, $z_i(\theta^M) = z(t_i; \theta^M) = (C_1(t_i; \theta^M), C_2(t_i; \theta^M))$ from Equations 38 and 39, meaning at each t_i we have an observation of both $C_1(t)$ and $C_2(t)$, $y_i^o = (C_{1,i}^o, C_{2,i}^o)$, and \mathbb{I} is a 2×2 identity matrix.

This model uses real data, so no ‘true’ parameter values exist. Instead, the MLE values of parameters are used for coverage simulations $\hat{\theta} = (0.00293, 0.00315, 0.00164, 78.8, 0.289, 0.0293, 1.83)$. The corresponding lower and upper parameter bounds are $a = (0.0001, 0.0001, 0, 60, 0.01, 0.001, 0.1)$ and $b = (0.01, 0.01, 0.01, 90, 1, 1, 3)$; the lower bounds for all the parameters apart from δ were zero [2] but were increased slightly to increase stability. Observation times are $t_{1:I} = (0, 769, 1140, 1488, 1876, 2233, 2602, 2889, 3213, 3621, 4028)$. Smaller nuisance parameter bounds are used for univariate profiles, although they are wider than those used in Simpson et al. [2]: $a_{\text{nuisance},j} =$

$\max(a_j, \hat{\theta}_j \div 2.5), j \in 1, 2, \dots, 7$ and $b_{\text{nuisance},j} = \min(b_j, \hat{\theta}_j \times 2.5), j \in 1, 2, \dots, 7$.

Real observations, the MLE model trajectory and the MLE 95% population reference set under this parameterisation can be seen in Figure 52.

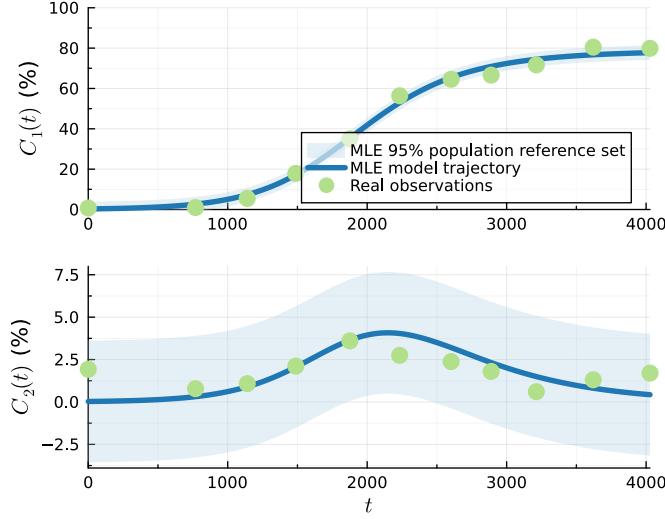


Figure 52: Two-species logistic MLE model trajectory, MLE 95% population reference set and real observations [2].

A.3.1 With Logit-Normal Data Distribution

When using a logit-normal data distribution instead of an additive Gaussian data distribution, the density function for the observed data becomes:

$$y_i \sim \text{LogitNormal}(\text{logit}(z_i(\theta^M)), \sigma^2 \mathbb{I}). \quad (41)$$

where $\theta^M = (\lambda_1, \lambda_2, K, \delta, C_1(0), C_2(0))$, $\theta^o = \sigma$ and $\text{logit}(p) = \log(p \div (1 - p))$. The model trajectory, $z_i(\theta^M)$, is assumed to be a proportion $\in (0, 1)$.

The ‘true’ parameter values we use here for coverage simulations are similar to the MLE values of the parameters when using this data distribution, albeit with a lower value of σ : $\theta = (0.003, 0.0004, 0.0004, 80.0, 0.4, 1.2, 0.1)$. Parameter bounds have been adjusted slightly to $a = (0.0005, 0.00001, 0.00001, 60, 0.01, 0.1, 0.01)$ and $b = (0.01, 0.005, 0.005, 98, 2, 3, 1)$ and no ‘special’ nuisance parameter bounds are specified.

For coverage testing of predictive quantities using the sampled full parameter confidence set, we use much more well-informed parameter bounds that may be overly constrained: $a_{\text{sampling}} = (0.0022, 0.00001, 0.0001, 73, 0.25, 0.7, 0.03)$ and $b_{\text{sampling}} = (0.0036, 0.001, 0.0009, 85, 0.65, 2, 0.2)$.

Example observations, the ‘true’ model trajectory and the 95% population reference set under this parameterisation can be seen in Figure 53.

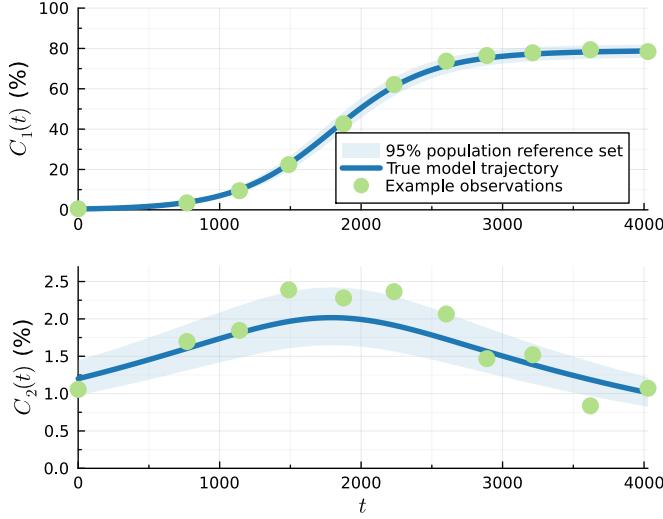


Figure 53: Two-species logistic model trajectory, 95% population reference set and example observations when using a logit-normal data distribution [2].

A.4 Gaussian Approximation of a Binomial Distribution

This model is taken from an unreleased paper by the authors of the workflow [1] and this thesis. The Binomial distribution is defined as:

$$X \sim \text{B}(n, p), \quad (42)$$

where \$n\$ is the number of trials and \$p\$ is the probability of success. For sufficiently large \$n\$, this distribution has the following Gaussian approximation:

$$y_i \sim p(y_i; \theta) X \sim \mathcal{N}(np, \sqrt{np(1-p)}), \quad (43)$$

where \$np\$ is the mean number of successes and \$\sqrt{np(1-p)}\$ is the standard deviation of this mean. We take the model parameter vector as \$\theta = (n, p)\$.

The true parameter values are \$\theta = (100, 0.2)\$. The corresponding lower and upper parameter bounds are \$a = (0.0001, 0.0001)\$ and \$b = (500, 1.0)\$. There is no observation time as such, but we take ten samples of the Gaussian approximation under the true parameterisation, \$y_{1:I}^0 = [21.9, 22.3, 12.8, 16.4, 16.4, 20.3, 16.2, 20.0, 19.7, 24.4]\$.

B Tolerance Set Coverage Example With Univariate Profiles

Here, we use the same example and data covered in Section 2.2.8.1 but show the profile-wise approximation of the full reference tolerance interval (see Figure 54). The univariate parameter profiles in Figures 54a and 54b contain their corresponding true values. The profile-wise reference tolerance interval formed by either univariate profile encloses the population reference interval in Figure 54e. Hence, the union of these profile-wise reference tolerance intervals also encloses the population reference interval.

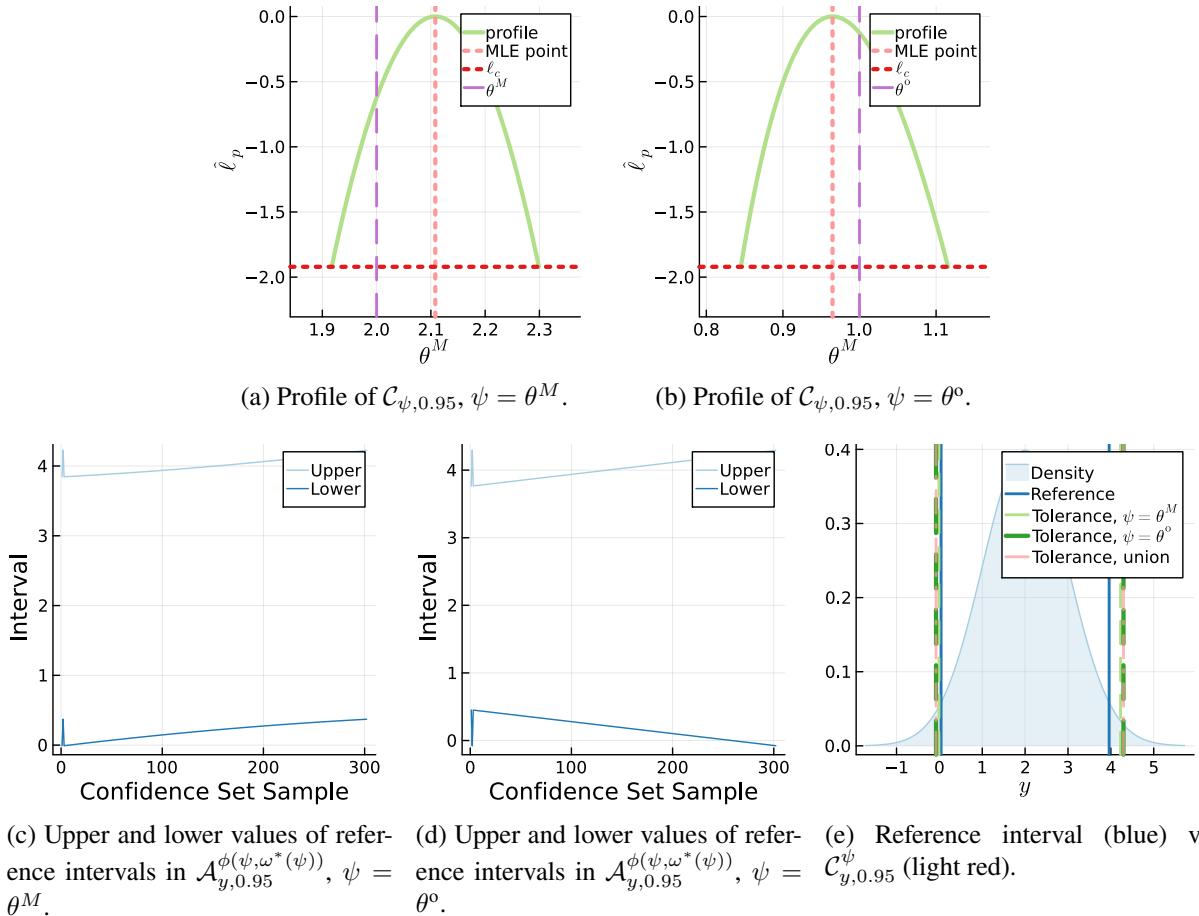


Figure 54: PWA workflow applied to obtain a profile-wise approximation of the full reference tolerance interval $\mathcal{C}_{y,(0.95,0.95)}$, which encloses the 95% population reference interval, given 100 observations of y .

C Bracketing Method Choice

Table 8 provides the number of likelihood and optimisation calls, with their corresponding ranks, each method requires to find the confidence interval for each parameter in the three-parameter logistic model [1].

Parameter	Method	Likelihood		Optimisation	
		Calls	Rank	Calls	Rank
1	FalsePosition	1660	1	23	1
	Brent	1843	2	28	5
	A42	1925	3	26	2
	Chandrapatla	1926	4	29	6
	ITP	1950	5	26	2
	AlefeldPotraShi	2022	6	27	4
	Bisection	2760	7	40	8
	Ridders	2976	8	38	7
2	Brent	1527	1	19	1
	Chandrapatla	1684	2	20	2
	FalsePosition	1736	3	21	3
	A42	1796	4	22	4
	AlefeldPotraShi	1805	5	22	4
	ITP	2153	6	26	6
	Bisection	2677	7	33	7
	Ridders	3331	8	42	8
3	FalsePosition	1242	1	20	1
	Chandrapatla	1412	2	24	2
	Brent	1565	3	26	4
	ITP	1568	4	24	2
	Ridders	2138	5	32	5
	Bisection	2305	6	38	6
	AlefeldPotraShi	2370	7	38	6
	A42	2382	8	38	6

Table 8: Likelihood and optimisation call rank, likelihood calls, optimisation calls of bracketing methods in `Roots` on each parameter in the three-parameter logistic model [1]. The likelihood rank of a method is calculated on a per-parameter basis by considering the number of likelihood calls, with ties allowed. The optimisation rank is calculated in the same fashion. Corresponds to Table 1.

D Univariate Profiles

D.1 Lotka-Volterra

The four univariate profiles for the Lotka-Volterra model [1] can be seen in Figure 55. The univariate profile log-likelihood function is very regular, as seen for the bivariate profiles in Appendix H.1, with the profile log-likelihood function very well approximated by the ellipse approximation.

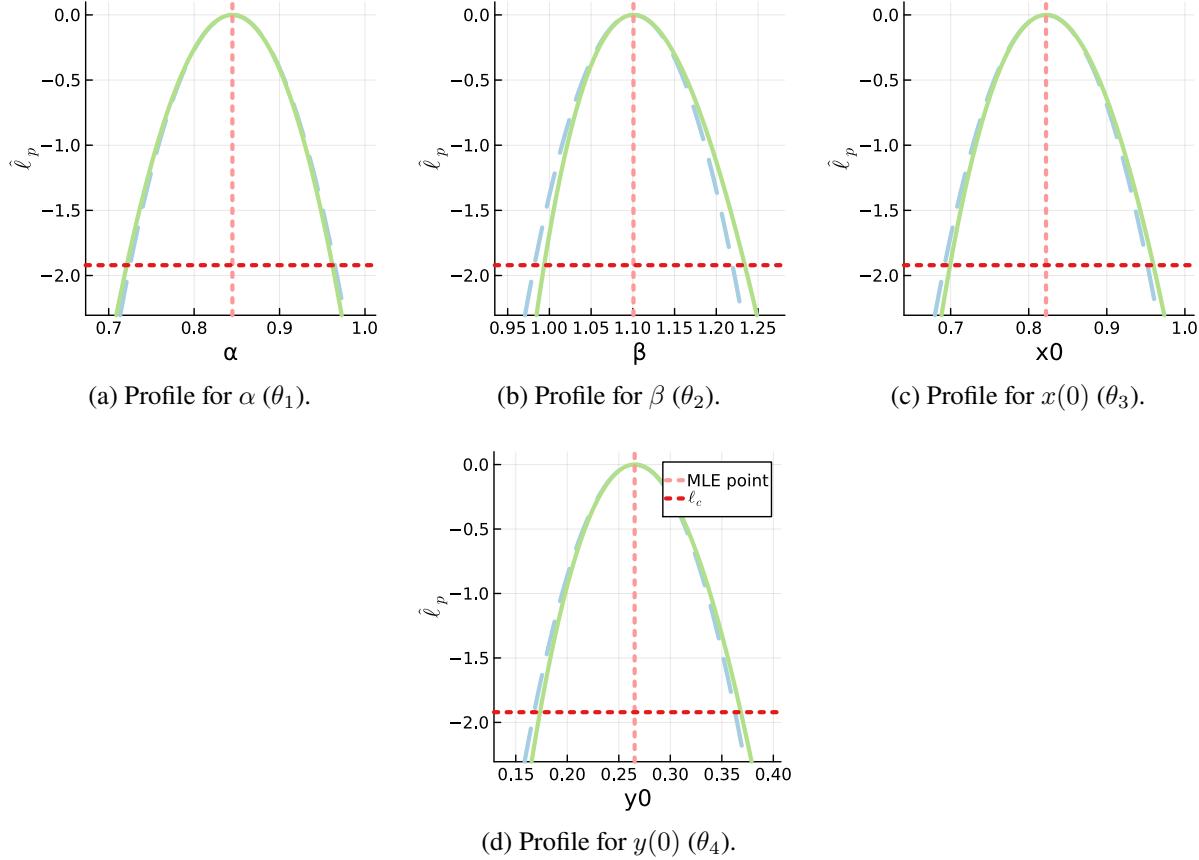


Figure 55: Normalised profile log-likelihood function (solid green) and ellipse approximation of that function (dashed light blue) for each parameter in the Lotka-Volterra model [1]. The vertical line (dotted light red) is the MLE value of each parameter. The horizontal line (dotted dark red) is the value of ℓ_c that corresponds to 95% likelihood-based confidence intervals for each parameter (intersection with the green line; intersection with the dashed blue line is the Wald confidence interval).

D.2 Two-Species Logistic

The seven univariate profiles for the two-species logistic model can be seen in Figures 56 (with smaller nuisance parameter bounds - see Appendix A.3) and 57 (nuisance parameter bounds equal to regular parameter bounds). Four of the profiles are consistent between both figures ($\lambda_1, K, C_1(0), \sigma$), with their log-likelihood based profiles also relatively well approximated by the ellipse approximation.

The profile for λ_2 is much wider without the smaller nuisance parameter bounds, but remains identifi-

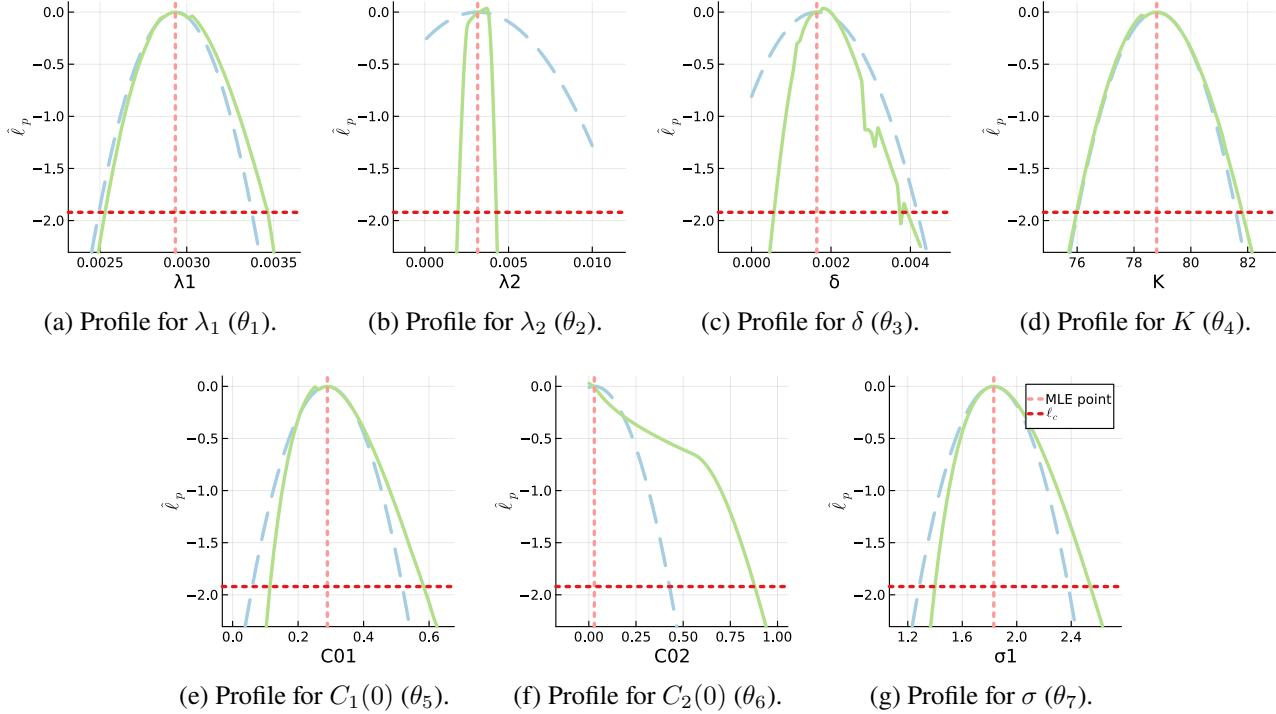


Figure 56: Normalised profile log-likelihood function (solid green) and ellipse approximation of that function (dashed light blue) for each parameter in the logistic two-species example [2], with smaller nuisance parameter bounds. The vertical line (dotted light red) is the MLE value of each parameter. The horizontal line (dotted dark red) is the value of ℓ_c that corresponds to 95% likelihood-based confidence intervals for each parameter (intersection with the green line; intersection with the dashed blue line is the Wald confidence interval).

able at a 95% confidence level. The profile for δ becomes much more spiky without the smaller nuisance parameter bounds, indicating issues with optimisation convergence in the profile log-likelihood function for this interest parameter. The δ term is an interaction term between the two coral subpopulations featured in the second differential equation as: $-\delta \times C_2(t) \times C_1(t) \div K$. It may be very hard to separate the impact of δ from these other parameters, particularly with limited data (there are only 22 observations, measured at 11 times), hence optimisation convergence may be very difficult for this parameter. This is further emphasised by the fact that the profile log-likelihood function for δ and λ_2 have sections just above 0.0, in both Figures, indicating that the actual maximum likelihood estimate was not found.

Finally, the profile for $C_2(0)$ is practically non-identifiable at a 95% confidence level without the smaller nuisance parameter bounds, while it is identifiable on the right hand side with the nuisance parameter bounds. This is in strong contrast to the result in Simpson et al. [2], which uses even tighter nuisance parameter bounds than used here; they found that $C_2(0)$ was identifiable with a confidence interval of [0.01, 0.29].

These results indicate that more data is needed to make all parameters identifiable without introducing tighter nuisance parameter bounds.

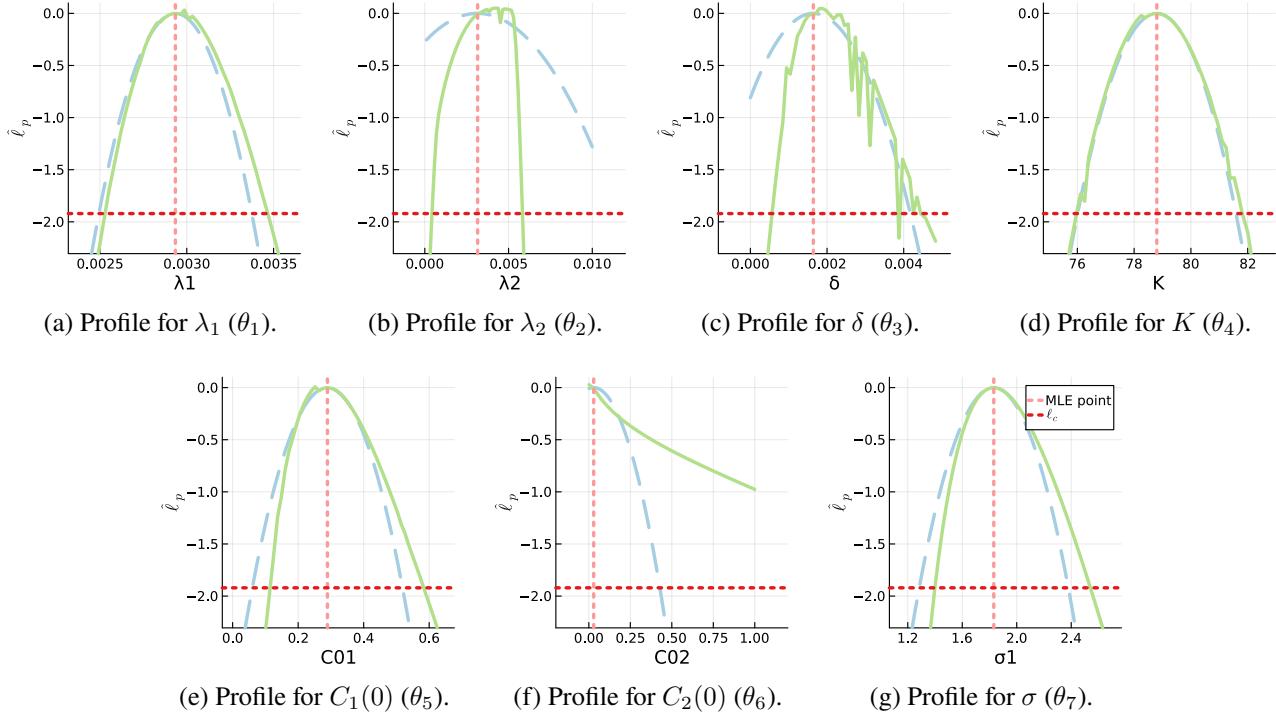


Figure 57: Equivalent output to Figure 56, but with nuisance parameter bounds that are the same as regular parameter bounds.

D.2.1 Logit-Normal Data Distribution

The seven univariate profiles for the two-species logistic model with a logit-normal data distribution can be seen in Figure 58. There are at least three important differences with the profiles here relative to those generated under the Gaussian model. Firstly, there are no signs of convergence issues on any profiles, and the ellipse approximation is a much more reasonable approximation in all the profiles. This contrasts several of the previous profiles, such as those for λ_2 and δ , which showed convergence issues and were not well approximated by the ellipse approximation. Secondly, $C_2(0)$ is now identifiable and has an MLE value above 1.0, whereas previously, it was closer to 0.0, which is more consistent with the observation of 1.927 at $t = 0$. Finally, λ_2 and δ are now non-identifiable on their left-hand side (i.e. could have a value of zero). This indicates that the coral growth for the second species could be explained by random noise in observations under this error model. Additionally, it indicates that the first and second coral species may not compete with each other.

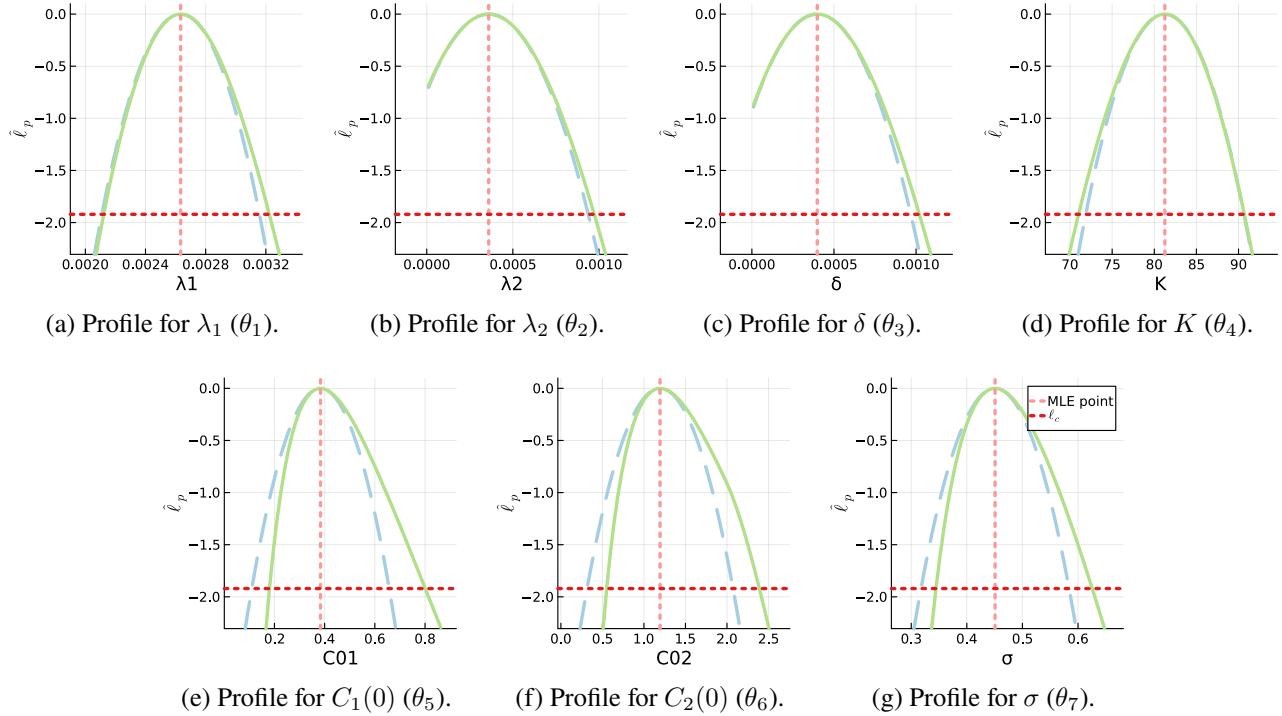


Figure 58: Normalised profile log-likelihood function (solid green) and ellipse approximation of that function (dashed light blue) for each parameter in the logistic two-species example [2], using a logit-normal distribution as the data distribution. The vertical line (dotted light red) is the MLE value of each parameter. The horizontal line (dotted dark red) is the value of ℓ_c that corresponds to 95% likelihood-based confidence intervals for each parameter (intersection with the green line; intersection with the dashed blue line is the Wald confidence interval).

E Bivariate Confidence Boundary Algorithm Pseudocode

E.1 General Algorithm used by SimultaneousMethod, RadialRandomMethod and RadialMLEMethod

SimultaneousMethod, RadialRandomMethod and RadialMLEMethod each use Algorithm 3 to find points on the confidence boundary. Where they differ is the specific method they use to generate point pairs that bracket these boundary points.

Algorithm 3 General method for finding boundaries of the bivariate confidence set, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o)$, of parameters j and k , $\theta_j \cup \theta_k$, which has a continuous unimodal profile log-likelihood function. The point pairs are found using the point pair algorithm for either SimultaneousMethod (Algorithm 4), RadialRandomMethod (Algorithm 5) or RadialMLEMethod (Algorithm 7). This example sets the confidence level of interest to 95%. Values of ω are the optimal values of the nuisance parameters, $\omega^*(\psi)$, given the most recent evaluation of the profile log-likelihood function. `find_zero` searches for a zero along the 1D line between the two 2D points ψ_1 and ψ_2 (the method to do this is an implementation detail).

Define the partitioning of parameters into a bivariate interest parameter $\psi = \theta_j \cup \theta_k$, $j < k$ and remaining nuisance parameters ω , with corresponding normalised profile log-likelihood function $\hat{\ell}_p(\psi; y_{1:I}^o)$

Define $\hat{\theta}$ as the maximum likelihood estimate of θ

Initialise $\text{atol} = 1 \times 10^{-3}$

Initialise $\hat{\psi} = [\hat{\theta}_j, \hat{\theta}_k]$, $1 - \alpha = 0.95$, $\ell_c = -\frac{\Delta_{\nu,1-\alpha}}{2}$, $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \emptyset$

Initialise $\psi_x = \theta_j$, $\psi_y = \theta_k$, $\psi_{\text{bounds},x} = [a_j, b_j]$, $\psi_{\text{bounds},y} = [a_k, b_k]$

Define a new function which is zero at the threshold of interest $f(\psi) = \hat{\ell}_p(\psi; y_{1:I}^o) - \ell_c$

Initialise internalQueue, externalQueue using the selected point pair algorithm (Algorithm 4,

5 or 7) with currently defined variables

while !empty(internalQueue) **do**

$(\psi_1) = \text{pop}!(\text{internalQueue})$

$(\psi_2, \omega^*(\psi_2), f_{\psi_2}) = \text{pop}!(\text{externalQueue})$

if $f_{\psi_2} \geq 0$ **then**

$\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_2, \omega^*(\psi_2))$

else

$\psi_{1-\alpha} = \text{find_zero}(f(\psi), (\psi_1, \psi_2), \text{Roots.Brent}(); \text{atol} = \text{atol})$

 Evaluate $f(\psi_{1-\alpha})$ to obtain values of $\omega^*(\psi_{1-\alpha})$

$\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_{1-\alpha}, \omega^*(\psi_{1-\alpha}))$

end if

end while

E.1.1 SimultaneousMethod Point Pairs

Algorithm 4 SimultaneousMethod used to find point pairs for Algorithm 3.

```

Initialise internalQueue =  $\emptyset$ , externalQueue =  $\emptyset$ 
Define min_num_unique =  $\lceil \text{totalPoints} \times \text{minimum proportion unique} \rceil$ 
while length(internalQueue) < totalPoints and length(externalQueue) < totalPoints do
    if length(internalQueue) == 0 then
        push!(internalQueue,  $(\hat{\psi})$ )
    else
         $\psi \leftarrow \text{rand}(\psi_{\text{bounds},x}) \cup \text{rand}(\psi_{\text{bounds},y})$ 
         $f_\psi = f(\psi)$ 
        if  $f_\psi > 0$  then
            push!(internalQueue,  $(\psi)$ )
        else
            push!(externalQueue,  $(\psi, \omega^*(\psi), f_\psi)$ )
        end if
    end if
end while
while length(internalQueue) < min_num_unique do
     $\psi \leftarrow \text{rand}(\psi_{\text{bounds},x}) \cup \text{rand}(\psi_{\text{bounds},y})$ 
     $f_\psi = f(\psi)$ 
    if  $f_\psi > 0$  then
        push!(internalQueue,  $(\psi)$ )
    end if
end while
if length(internalQueue) < totalPoints then
    Push unique elements of internalQueue to internalQueue in order, until length(internalQueue)
        equals totalPoints
end if
while length(externalQueue) < min_num_unique do
     $\psi \leftarrow \text{rand}(\psi_{\text{bounds},x}) \cup \text{rand}(\psi_{\text{bounds},y})$ 
     $f_\psi = f(\psi)$ 
    if  $f_\psi < 0$  then
        push!(externalQueue,  $(\psi, \omega^*(\psi), f_\psi)$ )
    end if
end while
if length(externalQueue) < totalPoints then
    Push unique elements of externalQueue to externalQueue in order, until length(externalQueue)
        equals totalPoints
end if

```

E.1.2 RadialRandomMethod Point Pairs

Algorithm 5 RadialRandomMethod used to find point pairs for Algorithm 3.

```

Initialise internalQueue =  $\emptyset$ , externalQueue =  $\emptyset$ 
while length(internalQueue) < totalPoints do
    if length(internalQueue) == 0 and use  $\hat{\psi}$  as first internal point then
         $\psi_1 = \hat{\psi}$ 
    else
        while true do
             $\psi \leftarrow \text{rand}(\psi_{\text{bounds},x}) \cup \text{rand}(\psi_{\text{bounds},y})$ 
            if  $f(\psi) > 0$  then
                 $\psi_1 = \psi$ 
                break
            end if
        end while
    end if
    Generate vector direction queue, directionsQueue, to search for the boundary from point  $\psi$ 
    using Algorithm 6
    while length(internalQueue) < totalPoints and length(directionsQueue) > 0 do
        currentDirection  $\leftarrow \text{pop!}(\text{directionsQueue})$ 
        Find  $\psi_2$  by searching from  $\psi_1$  in currentDirection until a bound in  $\psi_{\text{bounds},x}$  or  $\psi_{\text{bounds},y}$  is
        reached
        push!(internalQueue,  $(\psi_1)$ )
         $f_{\psi_2} = f(\psi_2)$ 
        push!(externalQueue,  $(\psi_2, \omega^*(\psi_2), f_{\psi_2})$ )
    end while
end while

```

Algorithm 6 Generation of num_radial_directions radial directions to search in for Algorithm 5.
Converts radial directions to vector directions.

```

Initialise startPointShift  $\leftarrow \text{rand}(0,1) \times 2\pi \div \text{num\_radial\_directions}$ 
Initialise directionsQueue =  $\emptyset$ , spacingRadians =  $2\pi \div \text{num\_radial\_directions}$ ,
    currentDirection = 0 radians
if  $\psi_x$  magnitude or  $\psi_y$  magnitude are NaN or Inf then
    Initialise relativeMagnitude = 1
else
    Initialise relativeMagnitude =  $\psi_x$  magnitude  $\div \psi_y$  magnitude
end if
for  $i \leftarrow 1$  to num_radial_directions do
    directionVector = [relativeMagnitude  $\times \cos(\text{currentDirection})$ ,  $\sin(\text{currentDirection})$ ]
    push!(directionsQueue, currentDirection)
    currentDirection = currentDirection + spacingRadians
end for

```

E.1.3 RadialMLEMethod Point Pairs

Algorithm 7 RadialMLEMethod used to find point pairs for Algorithm 3.

```

Initialise internalQueue =  $\emptyset$ , externalQueue =  $\emptyset$ 
Sample totalPoints, ellipsePoints using Algorithm 8
for point in ellipsePoints do
    currentDirection = point -  $\hat{\psi}$ 
    Find  $\psi$  by searching from  $\hat{\psi}$  in currentDirection until a bound in  $\psi_{\text{bounds},x}$  or  $\psi_{\text{bounds},y}$  is
        reached
    if point is contained in interest parameter bounds,  $\psi_{\text{bounds},x}$  and  $\psi_{\text{bounds},y}$  then
         $f_e = f(\text{point})$ 
        if  $f_e \leq 0.0$  then
            push!(internalQueue, ( $\hat{\psi}$ ))
            push!(externalQueue, (point,  $\omega^*(\text{point})$ ,  $f_e$ ))
        else
            push!(internalQueue, point)
             $f_\psi = f(\psi)$ 
            push!(externalQueue, ( $\psi$ ,  $\omega^*(\psi)$ ,  $f_\psi$ ))
        end if
    else
        push!(internalQueue, ( $\hat{\psi}$ ))
         $f_\psi = f(\psi)$ 
        push!(externalQueue, ( $\psi$ ,  $\omega^*(\psi)$ ,  $f_\psi$ ))
    end if
end for

```

Algorithm 8 Sample of totalPoints, ellipsePoints on the ellipse approximation of $\hat{\ell}_p(\psi; y_{1:I}^o)$ for Algorithm 7, using *Julia* package **EllipseSampling** [4].

```

Initialise startPointShift  $\leftarrow \text{rand}(0,1)$  (can also be user-specified)
Initialise clustering argument sqrt_distortion  $\in [0, 1]$ , where 0 is the strongest clustering and
    1 is equal spacing with respect to arc length
if the hessian of  $\hat{\ell}(\psi; y_{1:I}^o)$  at  $\hat{\theta}$  is not computed then
    Compute  $\mathcal{H}(\hat{\theta})$  and it's inverse  $\Gamma(\hat{\theta})$  (use a pseudo-inverse if not invertible, throwing a warning)
end if
Sample totalPoints, ellipsePoints, on the ellipse approximation of  $\hat{\ell}_p(\psi; y_{1:I}^o)$  at confidence
    level  $1 - \alpha$  using EllipseSampling's function generate_N_clustered_points with
    arguments  $\Gamma(\hat{\theta})$ , parameter indices  $j$  and  $k$ ,  $\hat{\psi}$ , startPointShift and sqrt_distortion

```

E.1.4 IterativeBoundaryMethod Point Pairs

Algorithm 9 IterativeBoundaryMethod used to iteratively improve an initial polygon approximation of a bivariate confidence boundary.

Find an initial boundary by starting from the end of Algorithm 3 with `numberOfInitialPoints` used as `totalPoints`, with either Algorithm 5 or Algorithm 7 used to find point pairs, using only the MLE point, $\hat{\psi}$, as an internal point

Note: require `numberOfInitialPoints` ≥ 3

Create edges between adjacent vertices on the initial boundary

Define `relativeMagnitude` as in Algorithm 6:

if ψ_x magnitude **or** ψ_y magnitude are `Nan` **or** `Inf` **then**

 Initialise `relativeMagnitude` = 1

else

 Initialise `relativeMagnitude` = ψ_x magnitude \div ψ_y magnitude

end if

Calculate `internalAngles` and `edgeLengths`, the internal angle and edge length objectives, using Algorithm 10 and 11, respectively

while `currentNumberOfPoints` $<$ `totalPoints` **do**

`iterMax` = `min(totalPoints, currentNumberOfPoints + anglePointsPerIteration)`

while `currentNumberOfPoints` $<$ `iterMax` **do**

 Determine the vertex with the largest objective in `internalAngles`, `currentVertex`

 Determine which of the two edges adjacent to `currentVertex` is connected to the vertex with the largest objective in `internalAngles`, `candidateEdge`

 Place a candidate point at the midpoint of `candidateEdge` as a initial boundary point guess, ψ_{guess}

 Use initial point ψ_{guess} to try and find a new boundary point, ψ_{new} , using Algorithm 12

 Run Algorithm 13 to determine how to proceed based on whether ψ_{new} was found

end while

`iterMax` = `min(totalPoints, currentNumberOfPoints + edgePointsPerIteration)`

while `currentNumberOfPoints` $<$ `iterMax` **do**

 Determine the edge with the largest objective in `edgeLengths`, `candidateEdge`

 Place a candidate point at the midpoint of `candidateEdge` as a initial boundary point guess, ψ_{guess}

 Use initial point ψ_{guess} to try and find a new boundary point, ψ_{new} , using Algorithm 12

 Run Algorithm 13 to determine how to proceed based on whether ψ_{new} was found

end while

end while

Algorithm 10 Internal angle objective. Uses *Julia* package `AngleBetweenVectors` [125] to calculate angles between adjacent edges (connected by a vertex). If two adjacent edges form a straight boundary then the objective is 0.0 radians, whereas if the boundary has an internal angle of $\pi \div 4$ radians (45 deg) the objective is $\pi \times 3 \div 4$ (135 deg).

Let `internalAngles` be a datastructure containing the internal angle objective corresponding to each boundary vertex

for `currentVertex` in initial boundary vertices **do**

- Determine the two edges connected to `currentVertex`, `edge1` and `edge2`
- Create vectors, `vector1` and `vector2` that represent the direction of each edge, rescaling the x magnitude by dividing x values by `relativeMagnitude` such that both dimensions have roughly the same weight

`internalAngles[currentVertex] ← the angle between the two vectors, vector1 and vector2`

end for

Algorithm 11 Edge length objective.

Let `edgeLengths` be a datastructure containing the edge length objective corresponding to each edge in the boundary

for `candidateEdge` in initial boundary edges **do**

- `edgeLengths[candidateEdge] ← the length of the candidateEdge (the euclidean distance),`
- rescaling the x magnitude by dividing x values by `relativeMagnitude` such that both dimensions have roughly the same weight

end for

Algorithm 12 Finding ψ_{new} given candidate point ψ_{guess} .

Find candidateDirection normal and pushing out from candidateEdge, rescaling the x magnitude by multiplying x values by relativeMagnitude² such that both dimensions have roughly the same weight

if $f(\psi_{\text{guess}}) \approx 0$ **or** candidateEdge is on a bound in $\psi_{\text{bounds},x}$ or $\psi_{\text{bounds},y}$ **then**

$\psi_{\text{new}} = \psi_{\text{guess}}$

else if $f(\psi_{\text{guess}}) > 0$ **then**

Find ψ_{bound} by searching from ψ_{guess} in candidateDirection until a bound in $\psi_{\text{bounds},x}$ or $\psi_{\text{bounds},y}$ is reached

$f_{\psi_{\text{bound}}} = f(\psi_{\text{bound}})$

if $f_{\psi_{\text{bound}}} \geq 0$ **then**

$\psi_{\text{new}} = \psi_{\text{bound}}$, new point is on interest parameter bounds

else

$\psi_{\text{new}} = \text{find_zero}(f(\psi), (\psi_{\text{guess}}, \psi_{\text{bound}}), \text{Roots.ITP}(); \text{atol} = \text{atol})$

Evaluate $f(\psi_{\text{new}})$ to obtain values of $\omega^*(\psi_{\text{new}})$

end if

else

Find oppositeEdge in the polygon candidateEdge is on, that is intersected by the line with candidateDirection, passing through ψ_{guess}

Find the oppositeVertex on oppositeEdge that is closest to ψ_{guess} , subject to oppositeVertex not being a vertex on candidateEdge (minimum 3 points per explored polygon so this is ok). Calculate these euclidean distances as in Algorithm 11

Line search from ψ_{guess} towards oppositeVertex to find a point between the two where $f(\psi) \approx 0$

if Point doesn't exist between ψ_{guess} and oppositeVertex where $f(\psi) \approx 0$ **then**

Failed to find ψ_{new}

else

Found ψ_{new}

Evaluate $f(\psi_{\text{new}})$ to obtain values of $\omega^*(\psi_{\text{new}})$

end if

end if

Algorithm 13 Boundary updates given successfully finding or failing to find ψ_{new} .

```

if  $\psi_{\text{new}}$  was found then
    currentNumberOfPoints  $\leftarrow$  currentNumberOfPoints + 1
    Break candidateEdge and replace with edges from each vertex in candidateEdge to the new
    vertex,  $\psi_{\text{new}}$ 
    Update internalAngles and edgeLengths given the broken edges and new edges
     $\mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) = \mathcal{C}_{\psi,1-\alpha}(y_{1:I}^o) \cup (\psi_{\text{new}}, \omega^*(\psi_{\text{new}}))$ 
else
    Break candidateEdge and oppositeEdge and reconnect the vertices in each edge such that there
    are now multiple boundary polygons.
if Only one or two points on one of these boundary polygons then
    Display an info message as no additional points can be found from the method directly.
    Set internalAngles and edgeLengths to zero for the vertices in the boundary polygon
        with one or two points
    Update internalAngles and edgeLengths for the new edges in the boundary polygon
        with three or more points
else if The largest boundary polygon has less than three points then
    Display a warning message and terminate the algorithm, returning the boundary found
        up until then.
end if
end if

```

F Iterative Boundary Further Discussion

F.1 Search Direction For External Candidate Points

When our candidate point is outside the desired boundary, we search towards the closest vertex on the opposite edge, subject to the vertex not being on the candidate edge. This is to guarantee that our search direction will hit a boundary point.

For example, consider the case where the bivariate log-likelihood function is bimodal, with two distinct MLE points, and the 95% confidence boundary can be represented by an ellipse and circle centred at each MLE point. The ellipse is centred at $(0, 0)$, has major and minor radii of 3 and 1, respectively, and has an anticlockwise rotation of 0.6π of the major radius from the x axis. The circle is centred at $(5, 3)$ with a radius of 1. In the first iteration of this example, we have four points found by starting at the $(0, 0)$ MLE point, as seen in Figure 59.

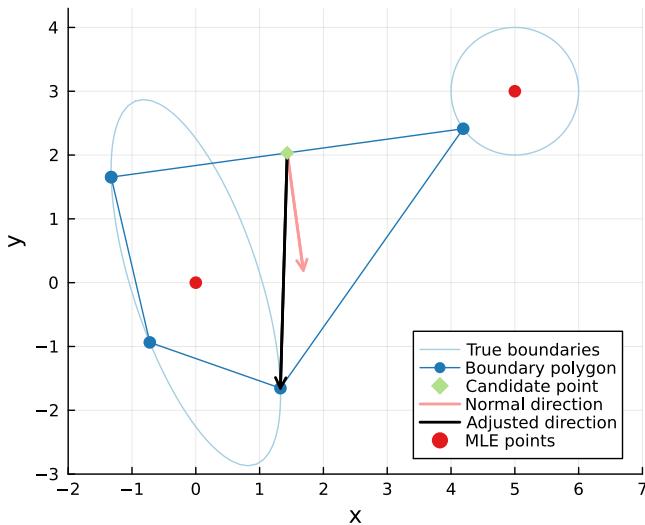


Figure 59: Search direction of an external candidate point in `IterativeBoundaryMethod` for a four point polygon with points on two separate boundaries.

Both the edge and angle objectives will select the topmost edge, placing a candidate point in the centre of it. This candidate point is an external point ($f(\psi) < 0$), and therefore we will search inwards. If we use the normal direction to search for a boundary point from the candidate point, it is clear that convergence of a line search method will be impossible. Instead, we adjust the search direction towards the closest vertex on the polygon edge intersected by the normal direction from the candidate point (and the vertex is not on the candidate edge). In that case, our line search can converge to a solution. This solution is not distinct from an existing polygon point, so the method enters the ‘failed to find’ boundary point stage. This allows for increased stability in the package’s implementation.

F.2 Relative Magnitude Scaling

When creating normal vectors from polygon edges, we need to rescale the x magnitude of the normal vector by `relativeMagnitude`² instead of only `relativeMagnitude` (Algorithm 12). This is because we want to rescale the parameter magnitudes such that they are approximately the same: this leads to a polygon in rescaled space (x magnitude is rescaled by `relativeMagnitude`). The normal vector from an edge of the polygon in rescaled space then has to be rescaled further by `relativeMagnitude` to bring it back into unscaled space.

For example, consider the case where the boundary we wish to find is given by two parameters, x and y , with magnitudes of 10 and 1, respectively, such that $\text{relativeMagnitude} = 10$. This example is shown in Figure 60. Let the confidence boundary be defined by a rectangle with points $\{[0, 0], [10, 0], [10, 1], [0, 1]\}$. Let the current boundary be defined by a triangle with the first three vertices of the rectangle. Let the current edge of the triangle be between points $[10, 1]$ and $[0, 0]$ with candidate midpoint $[5, 0.5]$. If we search in the normal direction to this edge in unscaled space, we will find the point $[4.95, 1]$ (normal vector is $[-1, 10]$). However, we would like to be able to find the other rectangle vertex, $[0, 1]$.

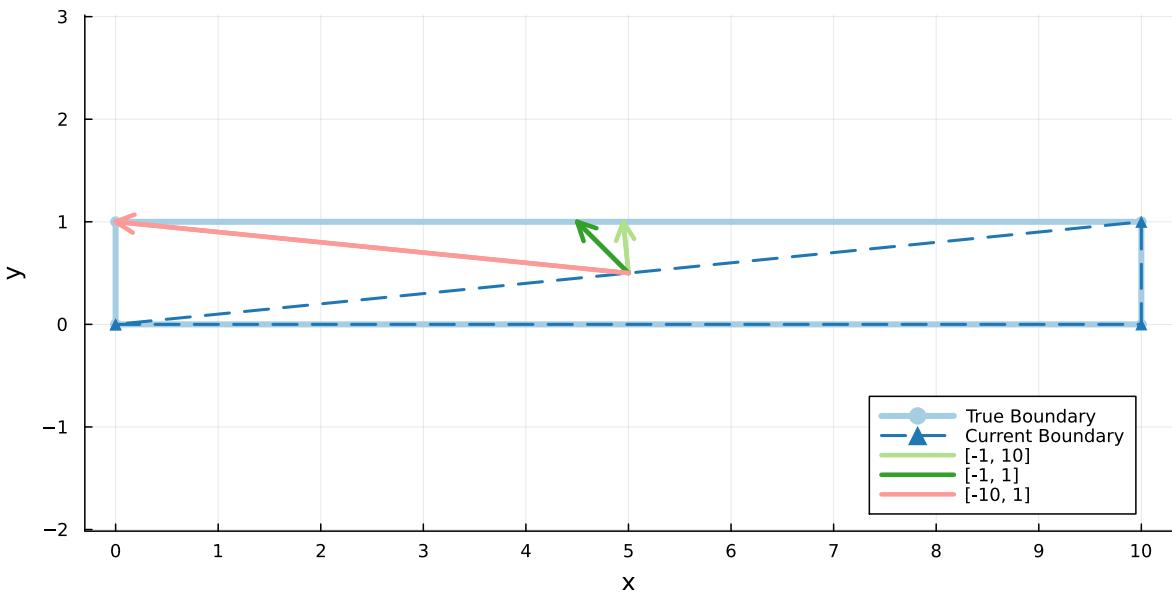


Figure 60: Impact of `relativeMagnitude` on rescaling confidence boundary search directions.

If we instead rescale the parameters by their relative magnitudes, then in scaled space, we have a rectangle with points $\{[0, 0], [1, 0], [1, 1], [0, 1]\}$. The normal direction to the equivalent edge is now $= [-10, 10] = [-1, 1]$, which finds the other rectangle vertex $[0, 1]$ from the scaled candidate midpoint $[0.5, 0.5]$. The normal direction in scaled space is thus the original direction with the x component divided by `relativeMagnitude`. However, if we try to use this normal direction without any additional scaling in unscaled space, we reach the point $[4.5, 1]$, which is again different from what we want.

Therefore, to obtain the equivalent normal direction in unscaled space, we must further divide the x component of the scaled normal direction by relativeMagnitude. I.e. we scale the original direction's x component by relativeMagnitude². If this is done, our normal direction is now $[-10, 1]$, which will find the desired vertex $[0, 1]$ from midpoint $[5, 0.5]$.

F.3 Demonstration on a Concave Example

Here, we demonstrate the `IterativeBoundaryMethod` on a highly concave 95% confidence log-likelihood boundary, as seen in Figure 61. The model used is the Gaussian approximation of a binomial distribution (Appendix A.4). The model uses synthetic data. As in Section 4.1.7's convex example, `IterativeBoundaryMethod` starts from the `RadialMLEMethod` solution with an initial three points, with no clustering, and a start point shift of 0.15. Another difference from the convex example is the presence of significantly different magnitudes for parameters n and p ; hence, the relative magnitude between them is set to 100. The aspect ratio of the plot is set to reflect how the relative magnitude parameter impacts the method. Angle and edge length objectives, as well as search directions, are all rescaled. This allows the objectives and directions considered to be directly observed from Figure 61.

The first five iterations find candidate midpoints inside the boundary; hence, their search direction is normal to each current edge. The sixth iteration instead finds a candidate midpoint outside the boundary. The search direction is, therefore, towards the closest vertex on the edge intersected by the line defined by the normal direction of the current edge, passing through the candidate midpoint.

The positive result of this method after 200 points is to effectively find the entire bivariate boundary, as seen in Figure 62. However, this is quite a large number of points and raises the question of the appropriateness of (effectively) a local search algorithm for this particular example. It is possible that other methods which feature more global exploration, like `RadialRandomMethod`, may be more appropriate. However, that still requires finding points within the small boundary relative to parameter bounds. This boundary could benefit greatly from an alternative approach like Bayesian LSE.

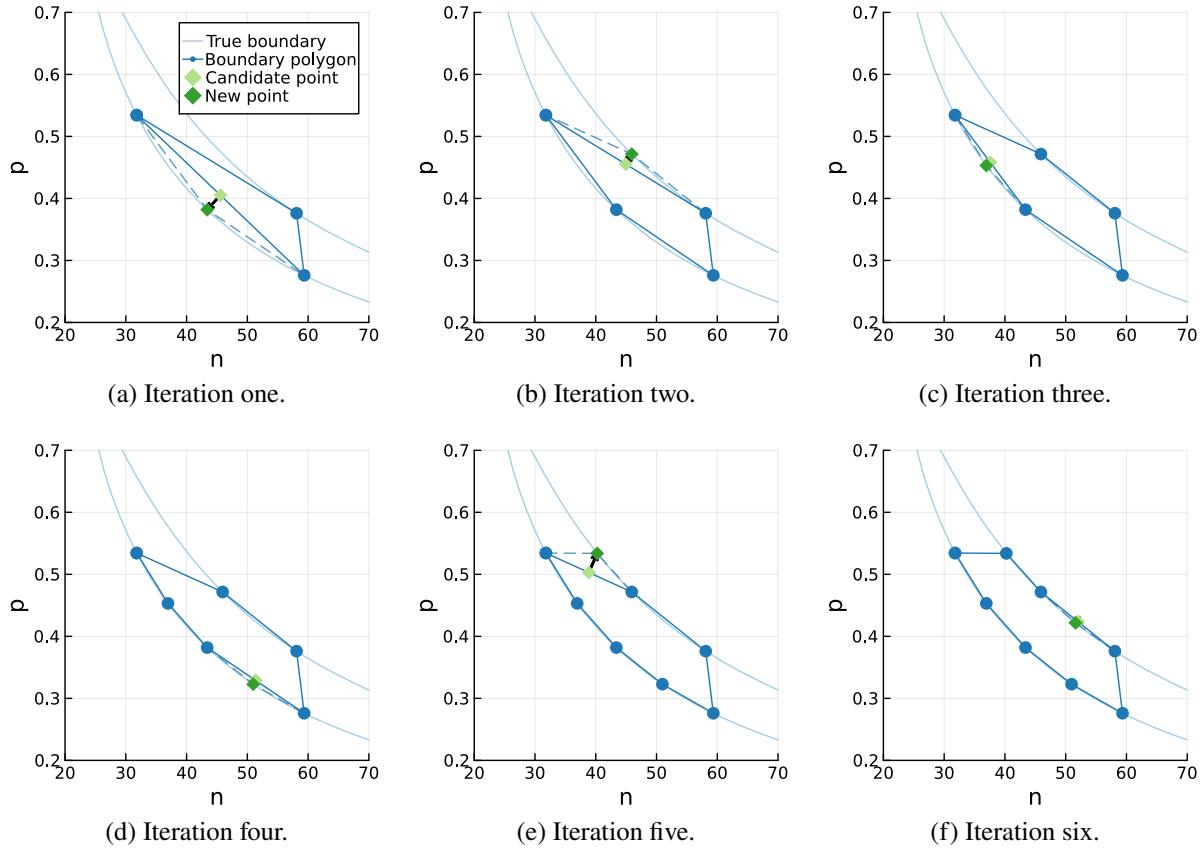


Figure 61: Six iterations of `IterativeBoundaryMethod` starting from a three point boundary found using `RadialMLEMethod`. Start point shift is set to 0.15 and equal spacing on the ellipse is used. The odd iterations consider the angle objective, while the even iterations consider the edge length objective. Dashed blue lines show the new edges of the polygon. Parameter magnitudes are set to 100 and 1 for n and p , respectively.

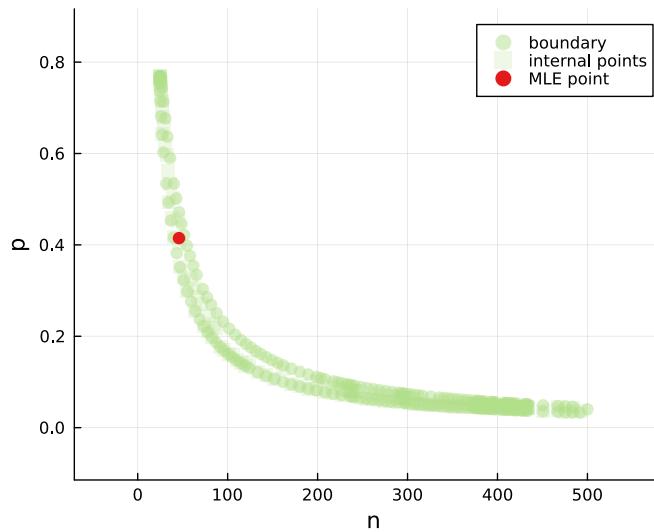


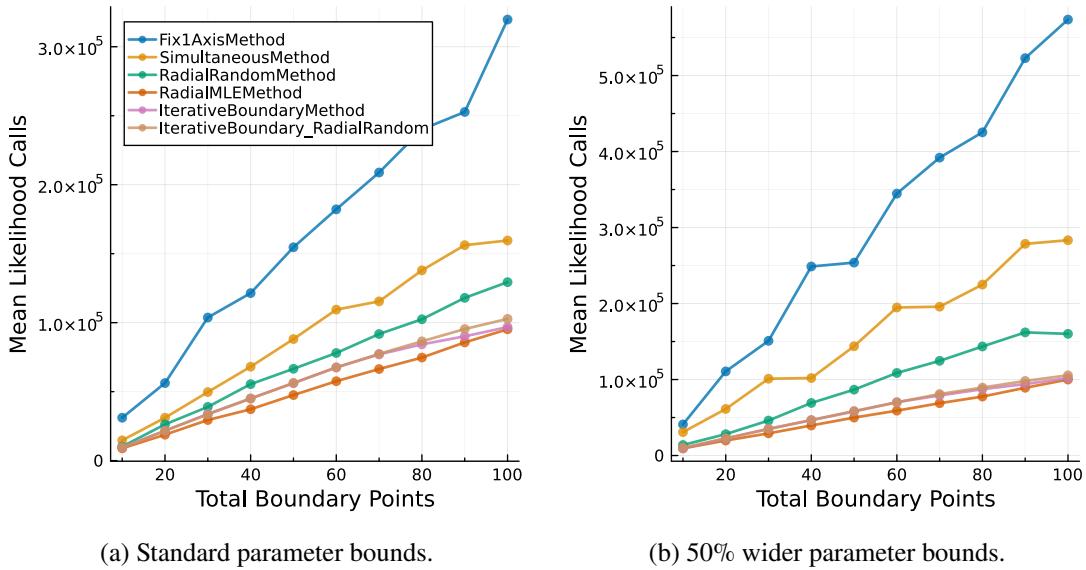
Figure 62: 200 point boundary from `IterativeBoundaryMethod` on the model from Figure 61.

G Bivariate Method Performance

G.1 Logistic

G.1.1 Optimisation and Likelihood Function Call Performance

The mean likelihood function call graphs in Figure 63 show effectively identical scaling behaviour to their corresponding mean optimisation function call graphs in Section 4.3.1. This demonstrates that only the optimisation function call graphs are required for comparison within these methods on this model.



(a) Standard parameter bounds.

(b) 50% wider parameter bounds.

Figure 63: Mean number of likelihood calls required to find a desired number of boundary points on each of the bivariate profiles in the logistic model [1] corresponding to the mean number of optimisation calls in Figure 21 for each method.

The number of optimisation calls for each distinct bivariate profile can be seen in Figure 64. Similar approximately linear trends can be seen for each method, with some significant stochasticity for the **Fix1AxisMethod** in particular. The number of function calls for each number of boundary points is evaluated separately. Hence, the case where a lower number of points requires more calls, such as at 90 points for **Fix1Axis** in Figure 64b, is down to randomness with its rejection sampling method.

The same behaviour as the mean case is evident when considering each profile separately using Figure 64. **Fix1AxisMethod** is the worst performer, scaling with around two times worse performance than all other methods (i.e. two times the additional optimisation calls required for every ten additional boundary points found). **SimultaneousMethod** is a significant improvement on **Fix1AxisMethod** in optimisation call performance and is slightly beaten by **RadialRandomMethod**. The remaining methods then beat these to a small degree, with **RadialMLEMethod** consistently the best or equal best

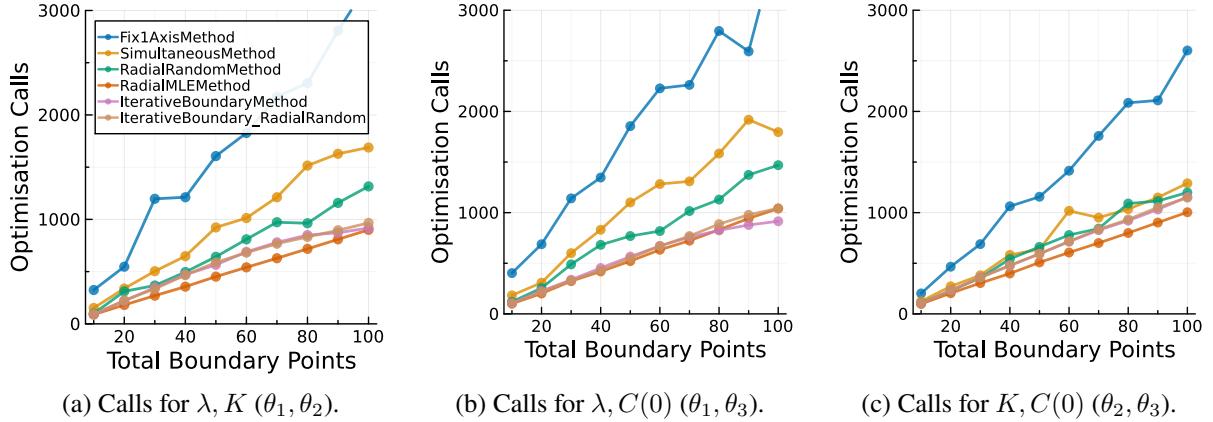


Figure 64: The number of optimisation calls required to find a desired number of boundary points on each of the bivariate profiles in the logistic model [1] corresponding to the mean number of optimisation calls in Figure 21 for each method.

performer. `RadialMLEMethod` narrowly beats out both of the `IterativeBoundaryMethod` settings, which show functionally identical performance to each other.

G.1.2 Coverage of True Bivariate 95% Confidence Set Boundary Using a Convex Hull

The area coverage of the true bivariate boundary found when using a convex hull to create the boundary polygon can be seen in Figure 64. These statistics correspond to the coverage graphs in Section 4.3.2, where an MPP method is used to find the polygon hull; all other simulation settings are kept fixed. The area coverage performance of the sampling-based `SimultaneousMethod` and `Fix1AxisMethod` is significantly better than when the MPP approach is used. `Fix1AxisMethod` now sees similar levels of coverage to `RadialRandomMethod` on the first two boundaries. The results for the remaining methods remain consistent, with `IterativeBoundaryMethod` the fastest to reach at least 98% area coverage on every boundary. Additionally, the performance statistics of `RadialMLEMethod` and `IterativeBoundaryMethod` using the convex hull are practically identical to the MPP statistics.

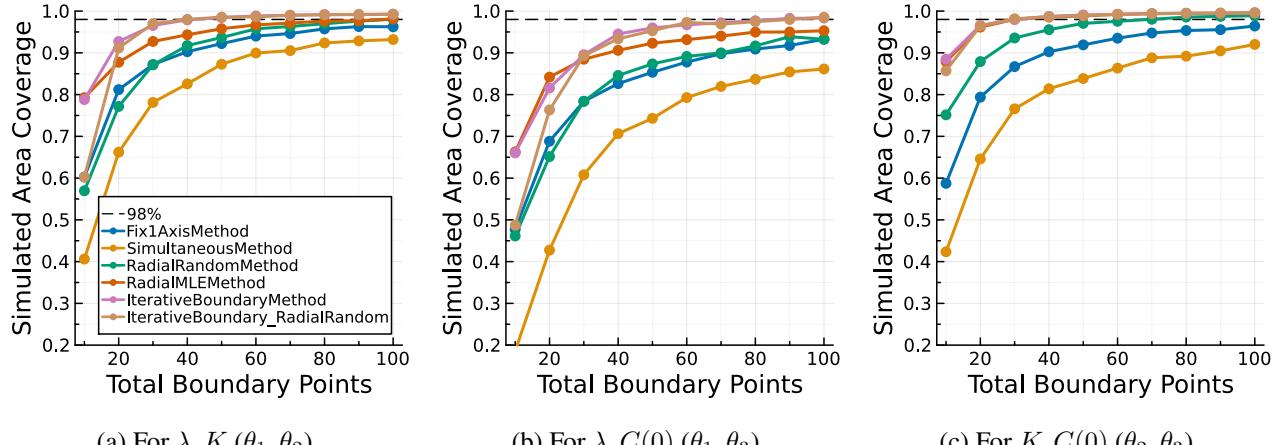


Figure 65: Approximate area coverage of the convex hull boundary polygon found from each method for increasing numbers of boundary points on each of the bivariate profiles in the logistic model [1] corresponding to the MPP coverage statistics in Figure 22.

H Bivariate Profiles

H.1 Lotka-Volterra

The six bivariate profiles for the Lotka-Volterra model [1] can be seen in Figure 55. The bivariate profile log-likelihood function is very regular, as seen for the univariate profiles in Appendix D.1, with the profile log-likelihood function very well approximated by the ellipse approximation.

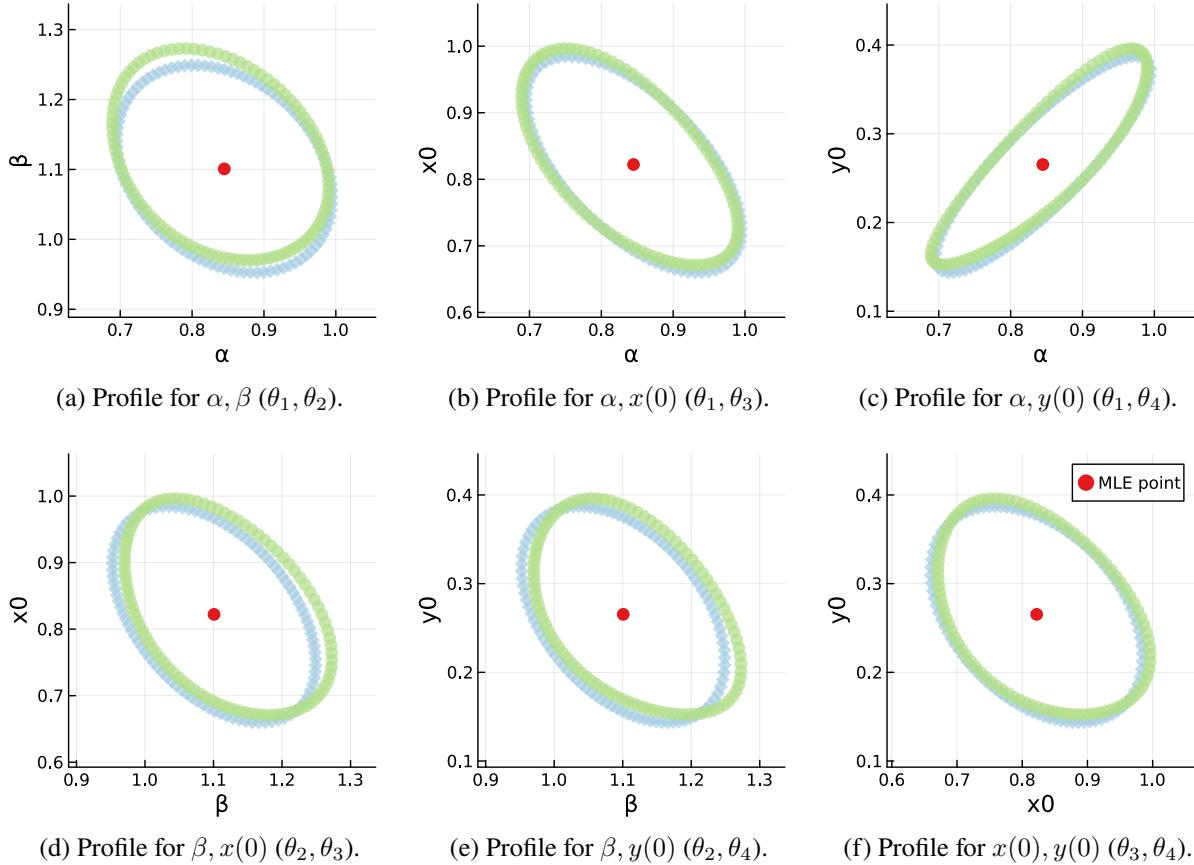


Figure 66: Points located on the true 95% confidence bivariate boundary (green dots) and ellipse approximation of the boundary (light blue diamonds) for each parameter pair in the Lotka-Volterra model [1].

H.2 Two-Species Logistic

The 21 bivariate profiles for the two-species logistic model with a logit-normal data distribution [2] can be seen in Figure 67.

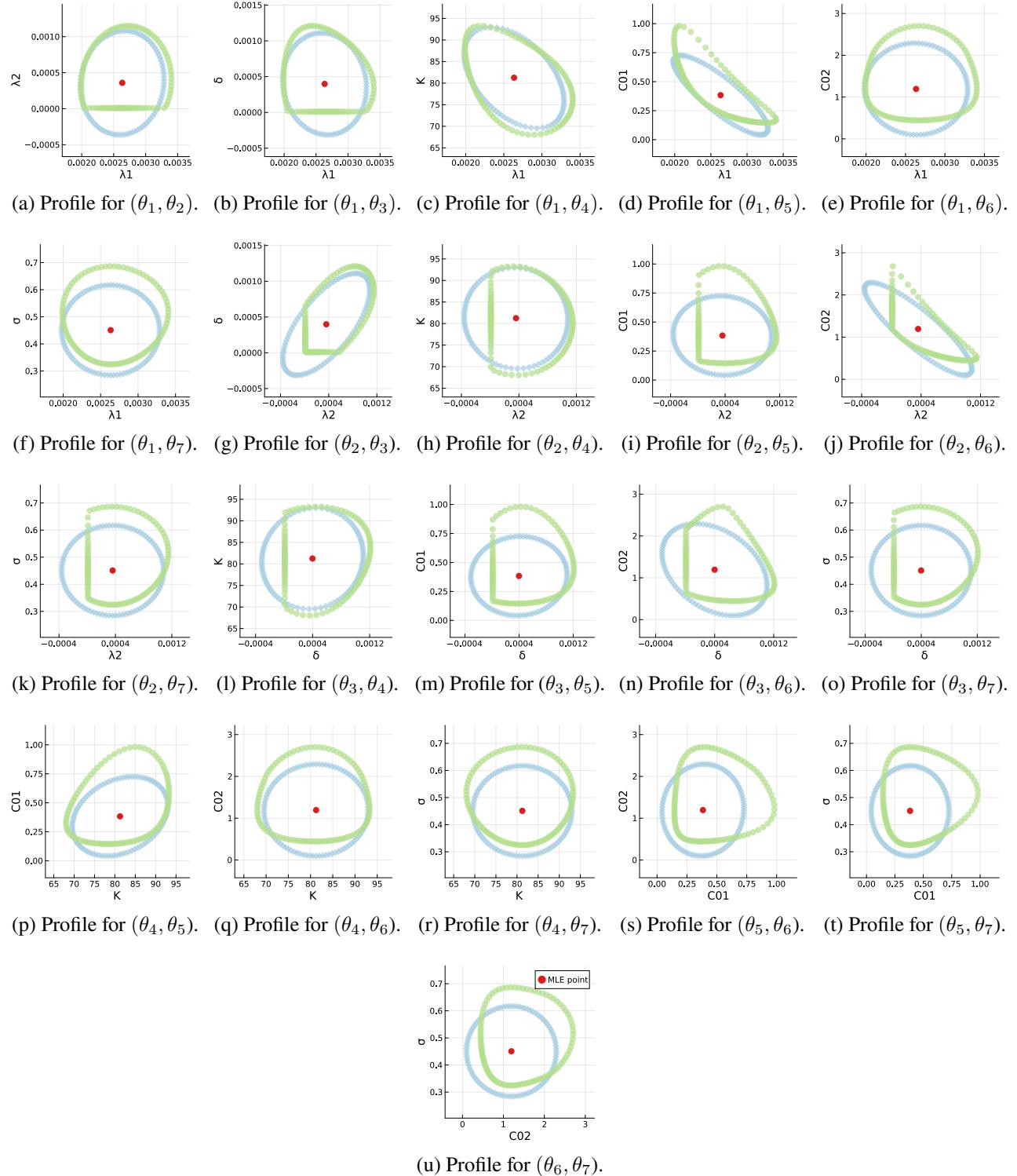
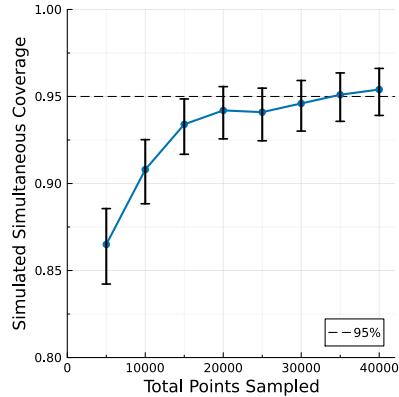


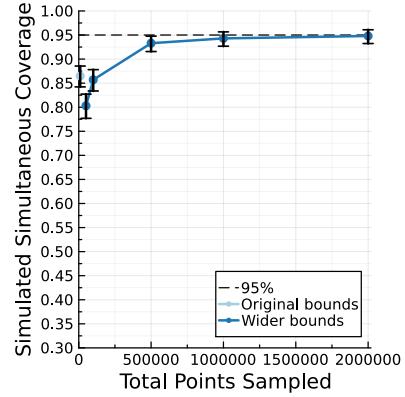
Figure 67: Points located on the true 95% confidence bivariate boundary (green dots) and ellipse approximation of the boundary (light blue diamonds) for each parameter pair in the two-species logistic model with logit-normal data distribution [2].

I Reference Tolerance Set Coverage

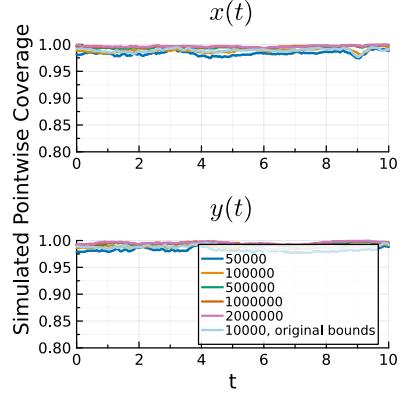
The full reference tolerance set coverage for the logistic and Lotka-Volterra models [1] can be seen in Figure 68. This figure corresponds effectively identically to the full trajectory confidence set coverage seen in Figure 27.



(a) Logistic simultaneous.



(b) Lotka-Volterra simultaneous.



(c) Lotka-Volterra pointwise.

Figure 68: Full reference tolerance set coverage in the logistic and Lotka-Volterra models [1] from 1000 coverage simulations. 95% confidence intervals for the simulated coverage are provided as error bars.

J Full Trajectory Confidence Set Coverage vs Full Reference Tolerance Set Coverage When θ^o Is Estimated

Here, we test if the coverage of full reference tolerance sets being higher than that of full trajectory confidence sets is a systematic result in this workflow when the data distribution standard deviation parameter is estimated. To do this we repeated the simultaneous trajectory confidence set and reference tolerance set coverage checks on the logistic and Lotka-Volterra models, now letting $\theta^o = \sigma$ be estimated, and on the two-species logistic model with an increased number of observations (from 22 to 222) as in Sections 3.2.2.1 and 4.5.2. With the increased number of observations, we expect any coverage issues associated with the lack of observations and the asymptotic threshold to be resolved. Each model's parameter bounds are adjusted to enable relatively efficient sampling of the full parameter confidence set without overly constraining the range of possible values. We sample a number of points that is large enough to cause full trajectory confidence set coverage to converge to approximately 0.95: ten million points for the logistic model, five million points for the Lotka-Volterra model and two million points for the two-species logistic model.

The results of this investigation can be seen in Figure 69. As in Section 5.1, 1000 simulations are performed on the logistic and Lotka-Volterra models, while 200 simulations are performed on the two-species logistic model, hence the wider confidence intervals for this last model. Once coverage has converged, the coverage is typically slightly higher for the full reference tolerance set in the Lotka-Volterra and two-species logistic models. Similarly, it is slightly lower for the full reference tolerance set in the logistic model.

For the logistic and Lotka-Volterra models, the confidence intervals for both coverage checks overlap, and each contains the other's point estimate (at least with a million or more points for the Lotka-Volterra model). While the observed testing and training data are consistent across each respective iteration, the random Latin Hypercube scheme chosen in each iteration is inconsistent. Resultantly, this difference is not statistically significant in these two models. Nonetheless, it indicates that the parameter confidence set in the logistic model is, on average, slightly underestimating the true θ^o . Similarly, in the Lotka-Volterra model, it is slightly overestimated.

The coverage difference of each prediction set in the two-species logistic model is more significant: the confidence interval of [0.86, 0.97] for the coverage of the full trajectory confidence set at two million sampled points does not contain the point estimate of 0.98 for the full reference tolerance sets coverage (Figure 69c. Resultantly, the parameter confidence is, on average, overestimating the true θ^o . The more significant difference for this model may be explained by the different data distribution model used

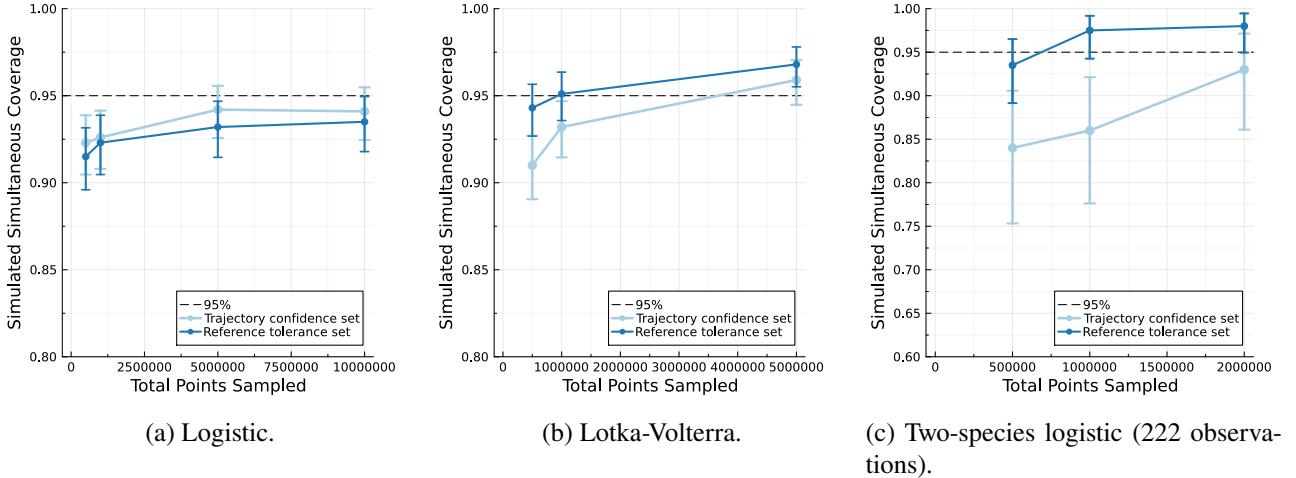


Figure 69: Simultaneous coverage of full trajectory confidence and full reference tolerance sets for increasing numbers of samples when $\theta^o \in \theta$. The logistic and Lotka-Volterra models used 1000 coverage simulations, while the two-species logistic model with more data used 200. 95% confidence intervals for the simulated coverage are provided as error bars.

(logit-normal instead of Gaussian) and its interaction with the estimated θ^o . However, $y_{1:I, \text{train}}^o$ is drawn from the data distribution used for each model, so this may not explain this difference. Additionally, it could be explained by the significantly higher number of observations relative to the other models (222 vs 11 and 42) coupled with the data distribution change. The two-species logistic model with fewer observations (22) had a difference in coverage that was more in line with the other two models (Section 5.1.2); the higher number of observations resolving issues with parameter coverage may be behind the more significant difference.

Figure 70 shows the impact of increasing the number of observations in the logistic and Lotka-Volterra models when θ^o is estimated on the simultaneous coverage of full trajectory confidence and full reference tolerance sets. The coverage of full trajectory confidence and full reference tolerance sets in both models for a sufficient number of points is *at least* 0.95 [1], with each a little lower or higher than the other. The point estimates of each set in the logistic model are contained within the confidence interval of the other set. Similarly, this also occurs in the Lotka-Volterra model for 100,000 and 200,000 observations. The behaviour is consistent with that observed in the previous examples.

Therefore, the result of full reference tolerance set coverage being higher than full trajectory confidence set coverage is not systematic within this workflow. Additionally, we would expect full trajectory confidence set coverage and full reference tolerance set coverage to be very similar when θ^o is estimated, although likely slightly different.

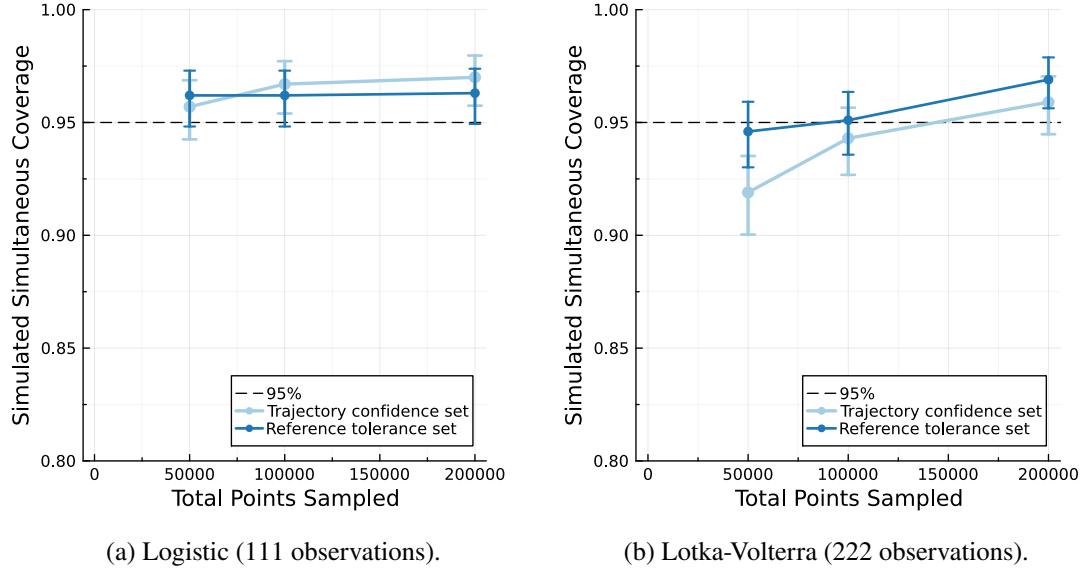


Figure 70: Simultaneous coverage of full trajectory confidence and full reference tolerance sets for increasing numbers of samples when $\theta^0 \in \theta$. The logistic and Lotka-Volterra models used 1000 coverage simulations and had an increased number of observations. 95% confidence intervals for the simulated coverage are provided as error bars.

K Profile Computational Performance Used for Final Comparisons

K.1 Univariate Profiles

Table 9, 10 and 11 give the mean number of likelihood calls required to find the confidence interval and any points within each interval for the logistic, Lotka-Volterra and two-species logistic models, respectively. The number of likelihood function calls for profiles formed using the asymptotic threshold at $\nu = 1$ and $\nu = |\theta|$ is considered.

Profile, ψ	Settings		Mean Calls	
	Total points in interval	$\log_{10}(\text{xtol_rel})$	$\nu = 1$	$\nu = \theta $
λ	20	-12	5880	6050
K	20	-12	6280	5900
$C(0)$	20	-12	17500	6100
			Total	29700 18100

Table 9: Performance of the method established in Section 3 in terms of the mean likelihood function calls (3 s.f.) required to find the confidence interval for each parameter across 100 data realisations on the logistic model [1]. $\nu = 1$ is at the univariate threshold, while $\nu = |\theta|$ is at the simultaneous threshold.

Profile, ψ	Settings		Mean Calls	
	Total points in interval	$\log_{10}(\text{xtol_rel})$	$\nu = 1$	$\nu = \theta $
α	0	-12	2770	2220
β	0	-12	2510	2220
$x(0)$	0	-12	2430	2010
$y(0)$	0	-12	2610	2460
			Total	10320 8910

Table 10: Performance of the method established in Section 3 in terms of the mean likelihood function calls (3 s.f.) required to find the confidence interval for each parameter across 100 data realisations on the Lotka-Volterra model [1]. $\nu = 1$ is at the univariate threshold, while $\nu = |\theta|$ is at the simultaneous threshold.

Profile, ψ	Settings		Mean Calls	
	Total points in interval	$\log_{10}(\text{xtol_rel})$	$\nu = 1$	$\nu = \theta $
λ_1	0	-12	6460	8650
λ_2	0	-12	8430	8800
δ	0	-12	8710	9490
K	0	-12	7420	9590
$C_1(0)$	0	-12	8400	9390
$C_2(0)$	0	-12	7130	8120
σ	0	-12	2610	2820
			Total	49200
				56900

Table 11: Performance of the method established in Section 3 in terms of the mean likelihood function calls (3 s.f.) required to find the confidence interval for each parameter across 100 data realisations on the two-species logistic model with a logit-normal data distribution [2]. $\nu = 1$ is at the univariate threshold, while $\nu = |\theta|$ is at the simultaneous threshold.

K.2 Bivariate Profiles

Table 12, 13 and 14 give the mean number of likelihood calls required to find the bivariate profiles with a specified number of boundary points for the logistic, Lotka-Volterra and two-species logistic models, respectively. The number of likelihood function calls for profiles formed using the asymptotic threshold at $\nu = 2$ and $\nu = |\theta|$ is considered.

Profile, ψ	Settings		Mean Calls	
	Total boundary points	$\log_{10}(\text{xtol_rel})$	$\nu = 2$	$\nu = \theta $
λ, K	50	-12	19600	20800
$\lambda, C(0)$	50	-12	13800	14000
$K, C(0)$	50	-12	20000	19900
			Total	53400
				54700
λ, K	20	-8	-	5780
$\lambda, C(0)$	20	-8	-	3460
$K, C(0)$	20	-8	-	5350
			Total	-
				14600

Table 12: Performance of the method established in Section 4.5 in terms of the mean likelihood function calls (3 s.f.) required to find a specified number of points on the bivariate boundary across 100 data realisations on the logistic model [1]. $\nu = 2$ is at the bivariate threshold, while $\nu = |\theta|$ is at the simultaneous threshold.

Profile, ψ	Settings		Mean Calls	
	Total boundary points	$\log_{10}(\text{xtol_rel})$	$\nu = 2$	$\nu = \theta $
α, β	30	-12	22500	22600
$\alpha, x(0)$	30	-12	23500	23100
$\alpha, y(0)$	30	-12	21300	21400
$\beta, x(0)$	30	-12	23400	23600
$\beta, y(0)$	30	-12	23600	23900
$x(0), y(0)$	30	-12	23100	23100
			Total	137000
				138000
α, β	20	-8	-	7660
$\alpha, x(0)$	20	-8	-	7800
$\alpha, y(0)$	20	-8	-	8470
$\beta, x(0)$	20	-8	-	8410
$\beta, y(0)$	20	-8	-	8270
$x(0), y(0)$	20	-8	-	7670
			Total	48200

Table 13: Performance of the method established in Section 4.5 in terms of the mean likelihood function calls (3 s.f.) required to find a specified number of points on the bivariate boundary across 100 data realisations on the Lotka-Volterra model [1]. $\nu = 2$ is at the bivariate threshold, while $\nu = |\theta|$ is at the simultaneous threshold.

Profile, ψ	Settings		Mean Calls	
	Total boundary points	$\log_{10}(\text{xtol_rel})$	$\nu = 2$	$\nu = \theta $
θ_1, θ_2	30	-12	82000	94300
θ_1, θ_3	30	-12	93300	108000
θ_1, θ_4	30	-12	103000	122000
θ_1, θ_5	30	-12	82900	90300
θ_1, θ_6	30	-12	71500	88100
θ_1, θ_7	30	-12	74000	89300
θ_2, θ_3	30	-12	130000	135000
θ_2, θ_4	30	-12	121000	130000
θ_2, θ_5	30	-12	126000	139000
θ_2, θ_6	30	-12	109000	111000
θ_2, θ_7	30	-12	89800	92700
θ_3, θ_4	30	-12	126000	134000
θ_3, θ_5	30	-12	132000	146000
θ_3, θ_6	30	-12	105000	114000
θ_3, θ_7	30	-12	99500	105000
θ_4, θ_5	30	-12	122000	133000
θ_4, θ_6	30	-12	101000	107000
θ_4, θ_7	30	-12	77200	87400
θ_5, θ_6	30	-12	113000	125000
θ_5, θ_7	30	-12	102000	113000
θ_6, θ_7	30	-12	90600	92000
Total			2.15 million	2.36 million
θ_1, θ_2	10	-8	-	24600
θ_1, θ_3	10	-8	-	28800
θ_1, θ_4	10	-8	-	32500
θ_1, θ_5	10	-8	-	26300
θ_1, θ_6	10	-8	-	24300
θ_1, θ_7	10	-8	-	22300
θ_2, θ_3	10	-8	-	40900
θ_2, θ_4	10	-8	-	37400
θ_2, θ_5	10	-8	-	40700
θ_2, θ_6	10	-8	-	32300
θ_2, θ_7	10	-8	-	25900
θ_3, θ_4	10	-8	-	39800
θ_3, θ_5	10	-8	-	41600
θ_3, θ_6	10	-8	-	34800
θ_3, θ_7	10	-8	-	30400
θ_4, θ_5	10	-8	-	37900
θ_4, θ_6	10	-8	-	30500
θ_4, θ_7	10	-8	-	24700
θ_5, θ_6	10	-8	-	38600
θ_5, θ_7	10	-8	-	29800
θ_6, θ_7	10	-8	-	25300
Total			-	669000

Table 14: Performance of the method established in Section 4.5 in terms of the mean likelihood function calls (3 s.f.) required to find a specified number of points on the bivariate boundary across 100 data realisations on the two-species logistic model with a logit-normal data distribution [2]. $\nu = 2$ is at the bivariate threshold, while $\nu = |\theta|$ is at the simultaneous threshold.