Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Why your ODE fit fails, and five steps to fix it: a practical guide

Mohammad Jamhuri

m.jamhuri@mat.uin-malang.ac.id

Universitas Islam Negeri Maulana Malik Ibrahim https://orcid.org/0000-0002-4506-7669

Putri Wulan Agustina

Universitas Islam Negeri Maulana Malik Ibrahim https://orcid.org/0009-0007-3508-5701

Method Article

Keywords: parameter estimation, model calibration, ordinary differential equations, solver tolerance, profile likelihood, gradient matching

Posted Date: October 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11445722/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Why your ODE fit fails, and five steps to fix it: a practical guide

Mohammad Jamhuri [1*] and Putri Wulan Agustina [1]

[1*]Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 65144, Indonesia.

*Corresponding author(s). E-mail(s): m.jamhuri@mat.uin-malang.ac.id;
Contributing authors: putriwulantina08@gmail.com;

Abstract

Fitting an ordinary differential equation (ODE) model to data is routine in mathematical biology, yet first attempts often end with a poor fit or with estimates that change from one initial guess to the next. We give a five-step recipe that works with any ODE solver and least-squares routine: rescale the problem, tighten the solver tolerance, start from a gradient-matching estimate, check the residuals, and report intervals on the log scale. The step that matters most is easy to miss. At the default relative tolerance of $\mathbf{10^{-3}}$ in Python, MATLAB and Julia, the finite-difference Jacobian of the default explicit solver is up to 31% wrong, whatever the difference step. On a Lotka–Volterra problem, tightening the tolerance to $\mathbf{10^{-6}}$ raises the success rate from 30–36% to 58–68% across three data sets and two optimisers. In 400 trials on four textbook models, each step rescues a different model, and the full recipe reaches the best fit in every trial. For the 1978 boarding-school influenza outbreak, the residual check exposes a flaw of the SIR model; a two-stage model removes it and lowers $\boldsymbol{R_0}$ from 3.96 to 3.18 (95% interval 2.82–3.65). Each step is a rule with a number and a warning sign.

Keywords: parameter estimation, model calibration, ordinary differential equations, solver tolerance, profile likelihood, gradient matching

1 Introduction

Most models in mathematical biology contain parameters that cannot be measured directly: a transmission rate, a predation rate, an elimination constant. They are

estimated by choosing the values for which the solution of the ODE model matches the data best, usually in the least-squares sense. This step is taught in every modelling course and used in thousands of papers, and the quality of every later prediction depends on it.

The step is harder than it looks because a failed fit rarely announces itself. The optimiser stops after a few iterations and reports success, or two initial guesses return two different answers, and the modeller is left to guess whether the cause is the model, the data, the optimiser or the code. The naive approach, which runs a general-purpose least-squares routine with its default settings from one initial guess, hides three separate difficulties: parameters on scales that differ by orders of magnitude, a Jacobian that is computed from inexact ODE solutions, and a cost surface with several minima.

Experienced groups have collected the remedies. Raue et al. [1] list lessons from a decade of fitting models in systems biology, a protocol by Villaverde et al. [2] sets out the full calibration workflow, benchmark collections compare optimisers on realistic problems [3, 4], and dedicated tools such as Data2Dynamics [5] and pyPESTO [6] build these lessons in. These works address users of specialised software and large models; the present guide addresses the newcomer who fits a small model with a general-purpose solver. Tutorials in this journal and elsewhere explain the statistical side of fitting epidemic models [7, 8], identifiability and prediction [9, 10], measurement-error models [11], and model building from data [12]; the present guide complements them with the numerical side. The primer on nonlinear regression by O'Brien and Silcox [13] explains why likelihood-based intervals are preferable to Wald intervals and names compartmental models based on differential equations, computational algorithms and starting values as topics for further study. This guide takes up these three topics.

Many newcomers fit their first models with the general-purpose ODE solvers and least-squares routines of Python, MATLAB, R or Julia, and with the default settings of those routines. This guide is written for that reader. It packages the remedies into five steps, each with a concrete setting and a warning sign that appears when the step is skipped (Section 3), and it measures how much each step helps on four test models and two classic data sets. The central finding concerns a setting that is invisible in most code. We show that the default solver tolerance, rather than the optimiser or the difference step, is the main hidden cause of failed fits in our tests, because the finite-difference Jacobian inherits the solver error, which reaches 31% at the default relative tolerance of 10^{-3} used by SciPy [14], MATLAB [15] and DifferentialEquations.jl [16].

The contributions of this guide are:

- a five-step recipe stated with numerical settings and warning signs, applicable with any ODE solver and least-squares routine (Section 3);
- a measurement of the effect of the solver tolerance and the difference step on the Jacobian and on the success of the fit, across random seeds, solvers and optimisers (Sections 3.2 and 4);
- an ablation on four textbook models in which the full recipe reaches the best fit in all 400 trials, with gradient matching as the most efficient start for an oscillating model (Section 4);
- a worked analysis of two classic data sets, including intervals and a residual check that shows where each model needs refinement (Section 5).

The individual techniques are established; the contribution lies in their measured effect and in a form that a newcomer can apply directly. Section 2 shows the problem on one example, Section 6 discusses the results and their scope, and Section 7 answers common practical questions.

2 A fit that fails for a hidden reason

We write the model as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{p}), \qquad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}), \tag{1}$$

with state vector $\mathbf{x}$ and parameter vector $\mathbf{p}$, and we observe some function of the state, $\mathbf{y}_j \approx \mathbf{g}(\mathbf{x}(t_j), \mathbf{p})$, at times $t_1, \dots, t_n$. The least-squares estimate minimises the sum of squared residuals

$$\mathrm{SSE}(\mathbf{p}) = \sum_{j=1}^{n} \bigl\| \mathbf{y}_j - \mathbf{g}(\mathbf{x}(t_j; \mathbf{p}), \mathbf{p}) \bigr\|^2 = \|\mathbf{r}(\mathbf{p})\|^2. \tag{2}$$

Every evaluation of $\mathbf{r}(\mathbf{p})$ requires a numerical solution of (1). Gauss–Newton-type methods such as Levenberg–Marquardt or trust-region least squares also need the Jacobian $\mathbf{J} = \partial \mathbf{r} / \partial \mathbf{p}$, which general-purpose routines approximate by finite differences: each parameter is perturbed by a small step and the model is solved again. Throughout, bold lowercase letters denote vectors, bold capitals denote matrices, functions such as log and $|\cdot|$ act componentwise on vectors, and model variables keep the conventional names of their field, such as S, I and R_0.

Figure 1 shows the consequence on the Lotka–Volterra predator–prey model (Table 1) with 31 observations of both species and 2% noise. Both panels use the same data, the same initial guess and the same least-squares routine. With the default solver tolerance 10^{-3} the routine stops after 22 model evaluations at parameters that are up to 287% wrong, and the fitted curves are visibly out of phase with the data (Fig. 1a). With the tolerance 10^{-6} it stops after 8 evaluations with all parameters within 2.4% of the true values (Fig. 1b). Nothing in the output of the first run signals a problem: the routine reports successful termination.

3 The recipe

Figure 2 summarises the five steps. Steps 1–3 make the optimisation reliable, and Steps 4–5 tell the modeller whether the result can be trusted. Each subsection states what to do, why it helps, and what warning sign appears when the step is skipped.

3.1 Step 1: rescale parameters and residuals

Biological parameters are positive and often differ by orders of magnitude; in the pharmacokinetic model of Table 1, the volume is 10 and the rate constants are between 0.2 and 1.5. We therefore estimate $\boldsymbol{\theta} = \log \mathbf{p}$ instead of $\mathbf{p}$. This keeps every parameter

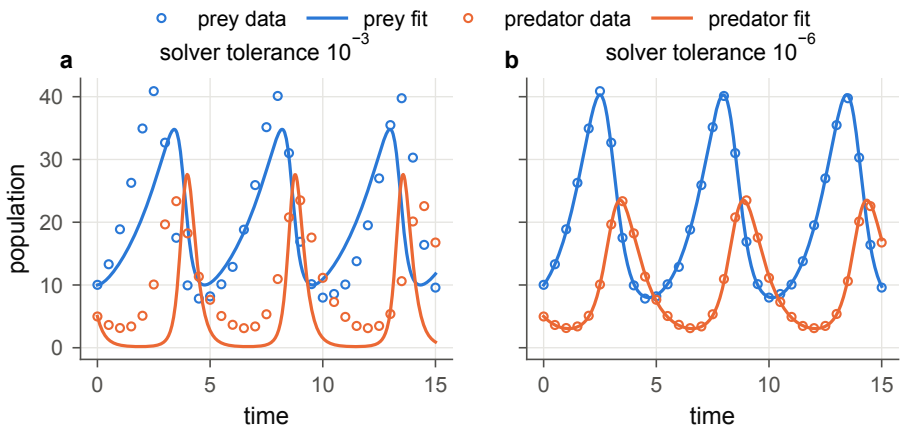


Fig. 1 The default solver tolerance alone can make a fit fail. Lotka–Volterra data (circles, 2% noise) and the fitted model (lines) from one initial guess with the relative solver tolerance (a) 10^{-3}, the default in several environments, and (b) 10^{-6}. In (a) the fit stops after 22 model evaluations with parameters up to 287% wrong; in (b) it stops after 8 evaluations with all parameters within 2.4% of the truth.

Step	What to do	Warning sign if skipped
1 Rescale	fit log-parameters; weight residuals by the noise	parameters differ by orders of magnitude
2 Tighten the solver	relative tolerance 10^{-6} (default is 10^{-3})	fit stops early or changes with the start
3 Start well	gradient matching if all states are observed, else 10 starts	different starts give different optima
4 Check residuals	plot residuals; lag-1 autocorrelation, runs	residuals show a pattern in time
5 Report uncertainty	profile-likelihood intervals for each parameter	profile is flat or open on one side

Fig. 2 The five-step recipe. The right-hand column lists the warning sign that typically appears when a step is skipped.

positive, makes a step of 0.1 mean "about 10%" for every parameter, and usually makes the cost surface closer to a quadratic bowl [17].

The residuals in (2) are scaled to match the noise. The choice of noise model matters for both the estimate and its uncertainty [11]. When the noise is roughly proportional to the signal, as for population counts or concentrations, we compare logarithms, $\log \mathbf{y}_j - \log \mathbf{g}_j$. For counts with Poisson-like noise, such as daily case numbers, we compare square roots, $\sqrt{\mathbf{y}_j} - \sqrt{\mathbf{g}_j}$. When several outputs are fitted together and no

noise model is known, each output is divided by its typical size so that no output dominates only because of its units.

Warning sign: estimates that hit zero or become negative, or one output that is fitted well while the other is ignored.

3.2 Step 2: tighten the solver tolerance

Adaptive ODE solvers choose their time steps so that the estimated local error stays below a tolerance of the form $\text{atol} + \text{rtol} \cdot |\mathbf{x}|$, with a relative tolerance rtol and an absolute tolerance atol. A relative tolerance of 10^{-3} means that the computed solution is correct to about three digits. This is adequate for plotting a solution, but a fit needs more, because the Jacobian is computed from *differences* of solutions. When the parameter is perturbed, the solver error changes as well, and the difference of two solutions contains the difference of two errors.

Figure 3 measures this effect on the Lotka–Volterra model. The relative error of the finite-difference Jacobian is about 31% at $\text{rtol} = 10^{-3}$, 0.6% at 10^{-4}, and 4×10^{-5} at 10^{-6}. It hardly depends on the difference step h over seven orders of magnitude, because the adaptive solver takes almost the same time steps for nearby parameter values, so the difference quotient differentiates the discrete solution and inherits its error. This is the internal differentiation effect described by Bock [18]. The practical rule is therefore: choose the solver tolerance, and leave the difference step at its default.

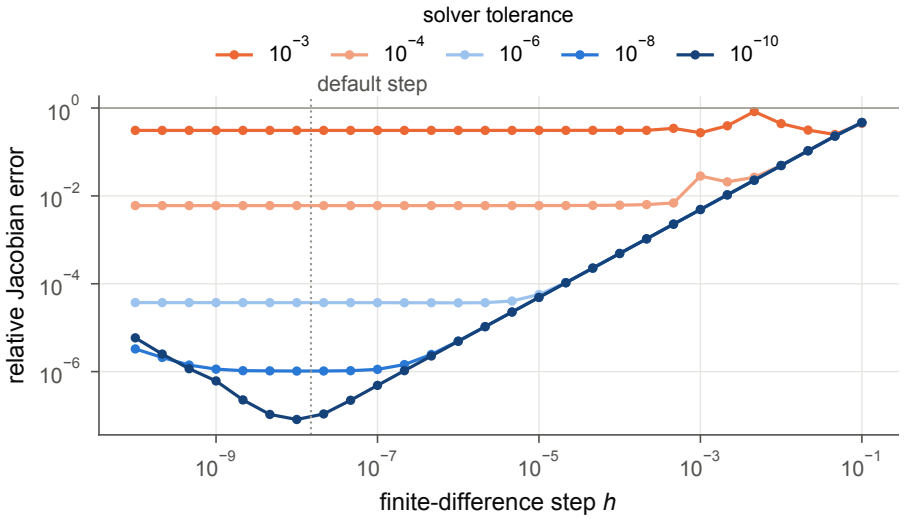


Fig. 3 The finite-difference Jacobian is only as accurate as the ODE solver. Relative error of the forward-difference Jacobian of the Lotka–Volterra model (log-parameters, at the true values) against the difference step h, for five solver tolerances. The dotted line marks the default step $\sqrt{\varepsilon} \approx 1.5 \times 10^{-8}$. The error is set by the tolerance and is nearly independent of h up to $h \approx 10^{-3}$.

Rule: set the relative tolerance to 10^{-6} and the absolute tolerance three orders of magnitude below the smallest state value of interest. On our test problems a fit

at 10^{-6} took about 1.4 times the wall time of a fit at 10^{-3}, because fewer iterations were needed (Section 4). The solvers of the R package deSolve [19] already use 10^{-6} by default. An alternative is to compute $\mathbf{J}$ from the sensitivity equations [20], which several dedicated tools do [5, 6]. For the Lotka–Volterra model, the Jacobian from sensitivity equations has an error of 0.5% at the default tolerance 10^{-3}, about 70 times smaller than the finite-difference error, and of 3×10^{-6} at 10^{-6}. Sensitivities therefore remove most of the problem; the tolerance rule remains useful because the solution itself is accurate to only about 1% at 10^{-3}.

Warning sign: the fit stops early with "success", or the estimate changes when the tolerance is changed. A quick test is to refit from the final estimate with a tolerance ten times smaller; a reliable estimate does not move.

3.3 Step 3: start close to the answer

Local optimisers find the minimum nearest to the initial guess, and ODE models, especially oscillating ones, can have several. Two remedies are cheap.

Gradient matching. When every state is observed, we can estimate the derivative $\dot{\mathbf{x}}$ directly from the data and avoid solving the ODE altogether [21–24]. Weak-form versions of the idea avoid differentiating noisy data and are an active topic in this journal [25, 26]. We fit a smoothing spline through each observed series, with the smoothing level set by the noise, and differentiate the spline. For many models the right-hand side is linear in the parameters, and the parameters then follow from a linear least-squares problem. For Lotka–Volterra with prey u and predator v,

$$\frac{\dot{u}}{u} = a - b\,v, \qquad \frac{\dot{v}}{v} = -c + d\,u, \tag{3}$$

so regressing the smoothed $\dot{u}/u$ on v gives (a, b), and regressing $\dot{v}/v$ on u gives (c, d). The result is used as the initial guess for the full least-squares fit.

Multistart. When some states are unobserved, or the model is not linear in its parameters, we run the fit from about ten initial guesses spread over a plausible range on the log scale and keep the best. For small models this costs seconds.

Warning sign: different initial guesses give different estimates whose SSE differs by more than 1%.

3.4 Step 4: check the residuals

A small SSE does not show that the model is adequate. We plot the scaled residuals against time. If the model captures the data, they look like noise: no trend, no waves, and about as many sign changes as a coin would produce. Two numbers summarise this. The lag-1 autocorrelation of the residuals should be below about $2/\sqrt{n}$, and the number of runs of equal sign should be near $n/2 + 1$. A systematic pattern indicates that the model misses a mechanism, or that the noise model of Step 1 is wrong, and it makes the uncertainty estimates of Step 5 too optimistic.

Warning sign: residuals that form waves or a trend, or a lag-1 autocorrelation above $2/\sqrt{n}$.

3.5 Step 5: report intervals on the log scale

The Jacobian at the optimum gives the usual approximate covariance $s^2(\mathbf{J}^\top\mathbf{J})^{-1}$, with $s^2 = \mathrm{SSE}/(m - n_p)$ for m residuals and n_p parameters. Computed for $\boldsymbol{\theta} = \log \mathbf{p}$, it gives the interval

$$p_i \in \left[\hat{p}_i\, e^{-q\,\sigma_i},\ \hat{p}_i\, e^{q\,\sigma_i}\right], \tag{4}$$

where σ_i is the square root of the ith diagonal element and q is the 97.5% quantile of the Student t distribution with $m - n_p$ degrees of freedom. On the log scale this interval is asymmetric in p_i, as it should be for a positive parameter.

The profile likelihood [10, 27–29] checks this interval without relying on the quadratic approximation. We fix θ_i at a sequence of values, refit all other parameters at each value, and record $\Delta(\theta_i) = m \log\bigl(\mathrm{SSE}(\theta_i)/\mathrm{SSE}_{\min}\bigr)$. The 95% interval contains the values where $\Delta \le m \log\bigl(1 + F_{1,m-n_p}/(m - n_p)\bigr)$, with $F_{1,m-n_p}$ the 95% quantile of the F distribution; this threshold matches the quantile q in (4) and tends to the familiar 3.84 for large m. A profile that stays flat on one side signals a parameter that the data cannot determine [28].

Warning sign: a profile that is flat or open on one side, or an interval that spans more than an order of magnitude.

4 How much each step helps

A recipe earns its place when each step makes a measurable difference. This section measures that difference on simulated data, where the true parameters are known: first the effect of the solver tolerance and the difference step, then its robustness across data sets, solvers and optimisers, and finally the gain from adding the steps one at a time.

4.1 Test problems

We tested the recipe on four models that a mathematical biologist meets early (Table 1): logistic growth, an SIR outbreak in which only the infected fraction is observed, a two-compartment pharmacokinetic (PK) model with oral dosing in which only the central concentration is observed, and the Lotka–Volterra model. Data were generated from the true parameters with multiplicative Gaussian noise. Appendix A gives the remaining details.

We call a fit successful when its SSE, recomputed with the relative tolerance 10^{-10}, is within 1% of the best SSE found for the same data set. This criterion judges each method by the minimum of its own objective and does not require the parameters to be identifiable. For the PK model, for example, the well-known "flip-flop" solution with k_a and k_{10} exchanged fits equally well and is counted as a success.

4.2 Solver tolerance and difference step

For the Lotka–Volterra and SIR models we ran 50 initial guesses on a grid of seven tolerances (10^{-3} to 10^{-10}) and six difference steps (Fig. 4). For Lotka–Volterra, all tolerances of 10^{-4} and below give the same result, 32–34 of 50 fits within 5% of the global optimum, at a median of 10–19 model evaluations. At 10^{-3} only 10–20 fits

Table 1 Test models: N is the population size with growth rate r and carrying capacity K; S and I are the susceptible and infected fractions; A_g, A_c and A_p are the drug amounts in the gut, central and peripheral compartments, with volume V and rate constants k_a, k_{10}, k_{12}, k_{21}; u and v are prey and predator. All parameters are estimated on the log scale; "obs." is the observed quantity and n the number of observation times.

Model	Equations	True parameters	Obs.	n
Logistic	$\dot{N} = rN(1 - N/K)$	$r = 0.8$, $K = 100$, $N_0 = 2$	N	25
SIR	$\dot{S} = -\beta SI$, $\dot{I} = \beta SI - \gamma I$	$\beta = 1.7$, $\gamma = 0.45$, $I_0 = 3/763$	I	15
PK	$\dot{A}_g = -k_a A_g$, $\dot{A}_c = k_a A_g - (k_{10} + k_{12})A_c + k_{21}A_p$, $\dot{A}_p = k_{12}A_c - k_{21}A_p$; dose 100	$k_a = 1.5$, $k_{10} = 0.2$, $k_{12} = 0.5$, $k_{21} = 0.3$, $V = 10$	A_c/V	12
LV	$\dot{u} = au - buv$, $\dot{v} = -cv + duv$	$a = 1$, $b = 0.1$, $c = 1.5$, $d = 0.075$; $u_0 = 10$, $v_0 = 5$	u, v	31

succeed, and they need 22–26 evaluations. The difference step changes these numbers little in any row, which confirms the message of Fig. 3. The SIR outbreak is short and does not oscillate. There, all 50 fits succeed at every grid point. The fits at 10^{-3} are nevertheless biased: their SSE is 0.3% above the minimum, and their parameters differ from it by 0.8%. The effect of the tolerance therefore depends on the model, and it is largest for oscillating systems, in which small errors in the period accumulate over time.

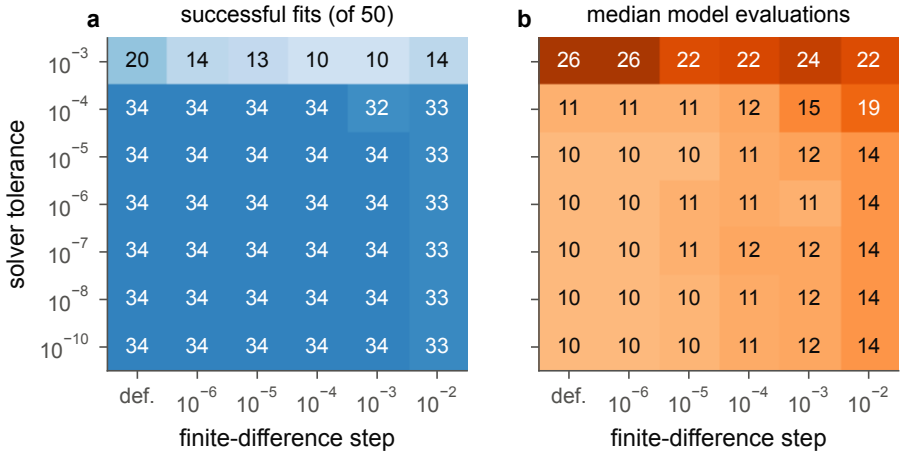


Fig. 4 Tolerance decides, difference step does not. Lotka–Volterra, 50 initial guesses per cell: (a) number of fits within 5% of the global optimum, (b) median number of model evaluations per fit. Columns are difference steps ("def." is the default $\approx 1.5 \times 10^{-8}$), rows are relative solver tolerances.

4.3 Other seeds, solvers and optimisers

The numbers above come from one data set and one solver. We therefore repeated the Lotka–Volterra experiment with three independent data sets, three solvers (the explicit Runge–Kutta method RK45, the automatic stiff/non-stiff switcher LSODA, and the implicit BDF method) and two optimisers (trust-region least squares and Levenberg–Marquardt), with 50 initial guesses each (Table 2). For the explicit Runge–Kutta solver, the type used by default in SciPy, MATLAB and Julia, tightening the tolerance from 10^{-3} to 10^{-6} raises the success count from 15–18 to 29–34 of 50 on every data set and with both optimisers, and reduces the median cost from 29 to 10 model evaluations with the trust-region method. For LSODA the gain is smaller (23–28 to 27–34), and BDF gives 28–33 successes already at 10^{-3}, at three to five times the computing time per fit. The Jacobian error explains the difference: at 10^{-3} it is 3% for BDF, 31% for RK45, 110% for LSODA and 170% for the eighth-order Runge–Kutta method DOP853, and at 10^{-6} it is below 0.2% for all four. At 10^{-6}, all six combinations of solver and optimiser reach success counts within two fits of each other on every data set. The rule of Step 2 therefore holds for every solver tested, and it matters most for the explicit methods that newcomers use by default.

Table 2 The tolerance rule holds across data sets, solvers and optimisers. Lotka–Volterra, three independent data sets with 50 initial guesses each; entries give the range over the data sets of the number of fits within 5% of the global optimum, and the median number of model evaluations. TRF: trust-region least squares; LM: Levenberg–Marquardt.

		rtol = 10^{-3}		rtol = 10^{-6}	
Solver	Optimiser	successes	evaluations	successes	evaluations
RK45	TRF	16–18	29	29–34	10
	LM	15–18	107	29–34	46
LSODA	TRF	23–28	25	27–34	11
	LM	25–27	88	28–34	54
BDF	TRF	29–33	28	29–34	10
	LM	28–30	93	29–33	46

4.4 Adding the steps one at a time

We then added the steps one at a time, with 100 trials per model; each trial used new noise (5%) and a new initial guess up to a factor $e \approx 2.7$ from the truth. S0 is the naive fit: original parameters, unweighted residuals, default tolerance and one initial guess. S1 adds Step 1, S2 adds Step 2, and S3 adds a ten-point multistart. For the two fully observed models we also ran gradient matching (GM) followed by a single fit with Steps 1–2.

Figure 5 shows that each step helps a different model. Logistic growth is the least demanding model, with 99 of 100 naive fits successful. Rescaling (S1) helps the PK model most, from 79 to 93 successes, because its parameters differ by two orders of

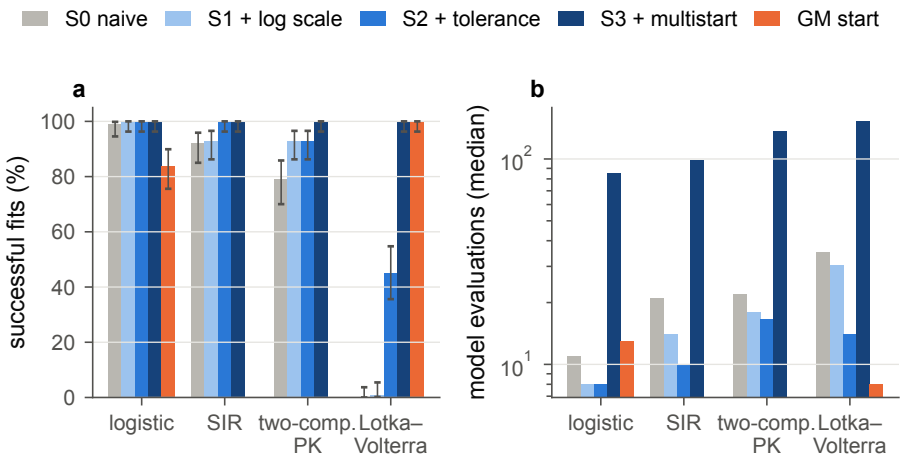


Fig. 5 Each step rescues a different model. Four models, 100 trials each, with the steps of the recipe added cumulatively: (a) percentage of successful fits with Wilson 95% intervals [30], (b) median number of model evaluations per fit (log scale; S3 counts all ten starts). GM is gradient matching followed by one fit, shown for the two fully observed models.

magnitude. The tighter tolerance (S2) completes the SIR model, from 93 to 100, and lifts Lotka–Volterra from 1 to 45. Multistart (S3) brings every model to 100 of 100, at about ten times the cost. Gradient matching is the most efficient start for Lotka–Volterra: 100 of 100 successes at a median of 8 model evaluations, fewer than a single fit from a random start. For logistic growth it succeeds in 84 trials, because a spline describes the early, nearly flat part of the curve poorly; there the multistart of S3 is the better choice. No single step suffices for all four models, and this is the reason to apply all of them.

5 Two classic data sets

Simulated data show how often each step helps; real data show what the recipe delivers in practice. We apply the full recipe to two data sets that are widely used in teaching, one from epidemiology and one from ecology, and follow all five steps through to the reported intervals.

5.1 Data and models

The first data set is the influenza outbreak in an English boarding school in January 1978 [31]: of 763 boys, the number confined to bed was recorded daily for 14 days. We use the series digitised by King and co-workers (Statements); another widely used digitisation differs by up to five cases on some days. We fit the SIR model $\dot{S} = -R_0\gamma SI/N$, $\dot{I} = R_0\gamma SI/N - \gamma I$ with $N = 763$, $S(0) = N - I_0$ and the observed number in bed equal to I. Its parameters are the basic reproduction number R_0, the recovery rate γ and the initial number of infected boys I_0, so that R_0 is estimated directly, and we compare square roots of the counts (Step 1). The second data set is the annual number of hare and lynx pelts traded by the Hudson Bay Company in

1900–1920 [32], a standard test of the Lotka–Volterra model [33]. We fit the four rates and the two initial populations h_0 (hare) and l_0 (lynx), six parameters in total, and compare logarithms.

5.2 Results

The recipe raises the success rate from random initial guesses for both data sets. For the influenza data, 34 of 50 naive fits reach the minimum of their own objective, and 42 of 50 fits with Steps 1–2 do so, at a median of 11 model evaluations against 22. For the lynx–hare data the numbers are 4 and 14 of 50, which shows that a six-parameter oscillating model needs Step 3. Gradient matching reaches the best fit in a single run with 9 model evaluations. The resulting estimates for the lynx–hare data, $a = 0.540$, $b = 0.0272$, $c = 0.796$ and $d = 0.0237$ per year, agree with a Bayesian analysis of the same data [33].

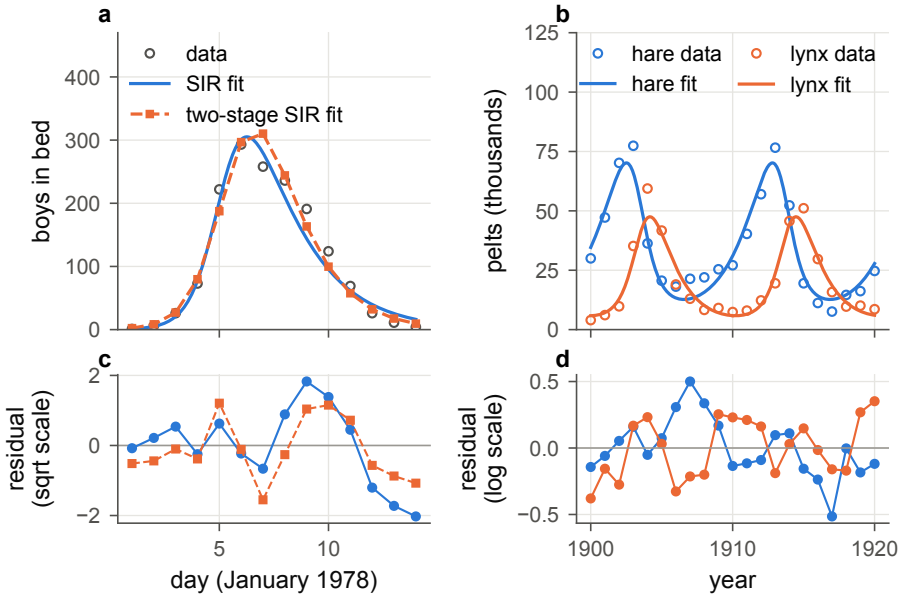


Fig. 6 The recipe fits two classic data sets, and the residuals show what each model misses. (a) Boys confined to bed during the 1978 boarding-school influenza outbreak ($N = 763$) with the SIR fit and the fit of an SIR model with two stages in bed; (b) Hudson Bay hare and lynx pelts, 1900–1920, and the Lotka–Volterra fit; (c, d) residuals on the scale used in the fit. The lag-1 autocorrelation of the residuals is 0.66 for the SIR fit and 0.34 for the two-stage fit of the influenza data, and 0.58 (hare) and 0.39 (lynx), against the thresholds $2/\sqrt{n} = 0.53$ and 0.44.

The fits look good at first sight (Fig. 6a, b), and the residual check of Step 4 adds a qualification. For the influenza data the residuals are small and of mixed sign until the peak, positive on days 8–11 and negative on days 12–14 (Fig. 6c), with a lag-1 autocorrelation of 0.66, above the threshold 0.53. The observed decline is first slower and then faster than the SIR model allows, a shape that a single exponential recovery

cannot produce. Avilov et al. [34] showed that the classic SEIR model also misses these data, and that a model with separate bed-confined and convalescent states and non-exponential residence times fits them; the residual pattern points to the same refinement without any additional analysis. For the hare series the autocorrelation is 0.58 against a threshold of 0.44, and the residuals change slowly from year to year, as expected for a two-species model of a system that involves more species. The intervals of Step 5 should therefore be read as lower bounds on the uncertainty.

The residual pattern also suggests a simple remedy. In the SIR model the time in bed is exponentially distributed, so most boys recover within a day or two and a few stay in bed for a long time. Splitting the bed-confined state into two consecutive stages, each left at rate 2γ, keeps the mean time in bed at $1/\gamma$ but makes its distribution peaked, which is the non-exponential residence time that Avilov et al. [34] found necessary. Fitted with the same recipe, this model has the same number of parameters, reduces the SSE from 16.1 to 9.7, lowers the lag-1 autocorrelation of the residuals from 0.66 to 0.34, below the threshold, and is preferred by the small-sample Akaike criterion (AICc 3.3 against 10.3). Three or four stages give similar fits, and adding a latent period instead (an SEIR model) lowers the SSE less (13.0) and leaves the residual pattern (0.55). The basic reproduction number of the two-stage model is $R_0 = 3.18$ (Jacobian interval 2.80–3.61, profile 2.82–3.65), with a mean time in bed of 2.1 days. This interval overlaps the SIR interval of Table 3 only between 3.30 and 3.65: the choice of model structure moves R_0 more than the noise in the data does, and the residual check is what reveals it.

Table 3 Estimates and 95% intervals for the two data sets: Jacobian interval on the log scale, Eq. (4), and profile-likelihood interval with the matching threshold.

Data	Parameter	Estimate	Jacobian interval	Profile interval
Influenza	R_0	3.96	3.23–4.85	3.30–4.89
	γ (day^{-1})	0.496	0.437–0.562	0.439–0.562
	I_0	1.16	0.34–3.93	0.30–3.28
Lynx–hare	a (year^{-1})	0.540	0.432–0.675	0.429–0.672
	b	0.0272	0.0204–0.0362	0.0203–0.0362
	c (year^{-1})	0.796	0.643–0.986	0.645–0.993
	d	0.0237	0.0179–0.0314	0.0180–0.0316
	h_0	34.6	29.1–41.1	29.4–40.8
	l_0	5.85	4.95–6.90	4.95–6.91

Table 3 and Fig. 7 compare the two kinds of interval. For eight of the nine parameters, the Jacobian interval on the log scale and the profile interval agree within a few per cent in width. For the number of initially infected boys, I_0, the profile is asymmetric and its upper limit is 3.3 instead of 3.9: the data constrain the early phase of the outbreak only weakly. This agrees with the caveat of O'Brien and Silcox [13] about Wald intervals: on the original scale the Jacobian interval for I_0 would even include negative values, and the log scale removes most, though not all, of the difference. The practical message is that the Jacobian interval on the log scale is a good first answer,

and that the profile is worth computing for parameters with wide intervals. Under the SIR model, the basic reproduction number of the outbreak is $R_0 = 3.96$ (3.30–4.89) and the mean time in bed is $1/\gamma = 2.0$ days.

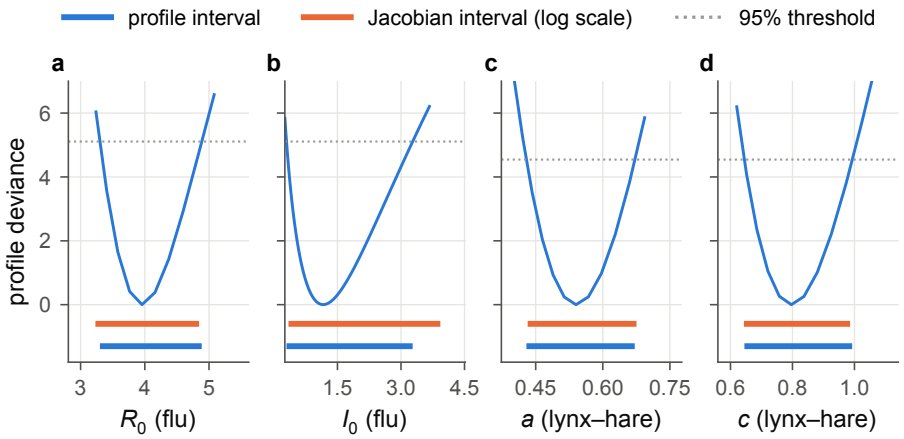


Fig. 7 On the log scale, Jacobian and profile-likelihood intervals agree, and the profile shows where they do not. Profile deviance for four parameters with the 95% threshold (dotted) and the two intervals. Only the weakly determined initial number of infected boys, I_0 (b), shows a clear difference.

6 Discussion

The five steps address three separate causes of failure, and the experiments show that each cause dominates for a different kind of model. Rescaling matters when parameters differ by orders of magnitude, as in the pharmacokinetic model. The solver tolerance matters when the solution oscillates, because a small error in the period grows over the time span and enters the finite-difference Jacobian. A good start matters when the cost surface has several minima, again most of all for oscillating models. A newcomer who meets only the first kind of model may conclude that the default settings are adequate, and then meet the second kind in a research project. Applying all five steps from the start removes this risk at a cost of seconds for small models.

The tolerance result agrees with the experience of groups that develop dedicated software, which compute the Jacobian from sensitivity equations and recommend tight tolerances [1, 5, 6]. Our measurements add two points that matter for users of general-purpose routines. First, the error of the finite-difference Jacobian is set by the solver tolerance and hardly by the difference step, so tuning the step brings little benefit. Second, the size of the error at the default tolerance depends on the solver, from 3% for BDF to 170% for DOP853, while a relative tolerance of 10^{-6} brings it below 0.2% for every solver we tested. A single rule therefore covers all of them.

The residual check of Step 4 turns the recipe from a tool for estimation into a tool for modelling. In both classic data sets the best fit of the textbook model leaves residuals with a clear time pattern, and this pattern points to the refinement that a later

study found necessary for the influenza outbreak, non-exponential residence times [34]. For the influenza data, a two-stage model with the same number of parameters removes the pattern and moves the estimate of R_0 from 3.96 to 3.18. For teaching, this is a useful sequence: fit the simple model reliably first, then let the residuals suggest the next model.

The recipe is designed for, and tested on, models with up to six parameters, deterministic ODEs, and observation noise that is additive on the scale chosen in Step 1. The test problems cover monotone, saturating and oscillating dynamics, full and partial observation, and noise levels of 2% and 5%. For models with tens of parameters, stiff systems with many states, or data from several experimental conditions, the same principles apply, and the dedicated tools discussed in Section 7 provide them with sensitivity-based derivatives. Each step is stated with a number and a warning sign, so it can be checked on a new model in the same way as in this guide.

7 Practical questions

My model is stiff. Does Step 2 still apply?

Yes. Stiff models need an implicit solver (BDF, Rosenbrock or LSODA-type methods), and the same tolerance rule applies. A fit that becomes 10 or more times slower with an explicit solver at tight tolerance is a strong hint of stiffness.

Some states are not observed. Can I use gradient matching?

Not directly, because the derivative of an unobserved state cannot be smoothed from data. Use the multistart of Step 3, or use gradient matching for the observed part of the model when the equations allow it.

The profile of one parameter is flat.

The data do not determine that parameter. Fix it from independent information, reparametrise the model (as we did with R_0), or design a new experiment; the tutorials in [9, 28] discuss these options.

When should I move to dedicated software or Bayesian methods?

For models with tens of parameters, many experimental conditions, or partial and censored observations, dedicated tools such as Data2Dynamics [5] and pyPESTO [6] provide sensitivity-based gradients, multistart and profiles in one package. Bayesian methods are the natural next step when prior information is available or when the full uncertainty distribution is needed [7, 33, 35]. The steps of this guide remain useful there, because a good starting point and well-scaled parameters help every method.

Exercises for a computer laboratory

The following exercises reproduce the main findings with any ODE solver and least-squares routine. Answers are given in Appendix B.

E1 (Step 2). Simulate the Lotka–Volterra model of Table 1 with 2% multiplicative noise. Fit it from 50 initial guesses drawn log-uniformly within a factor $e^{0.7}$ of the truth, once with relative tolerance 10^{-3} and once with 10^{-6}. Count the fits whose parameters lie within 5% of the best fit, and record the number of model evaluations.

E2 (Step 2). At the true parameters, compute the finite-difference Jacobian with tolerance 10^{-3} and steps 10^{-8}, 10^{-6} and 10^{-4}, and compare it with a Jacobian computed at tolerance 10^{-10}. Which setting changes the error, the step or the tolerance?

E3 (Step 3). For the lynx–hare data, smooth the logarithm of each series with a spline, differentiate it, and estimate a, b, c, d from the linear regressions (3). Use the result as the initial guess of a full fit and compare it with the final estimate.

E4 (Steps 4–5). Fit the SIR model to the influenza data and compute the lag-1 autocorrelation of the residuals. Then split the bed-confined state into two stages, each left at rate 2γ, refit, and compare the residuals, the AICc and the estimate of R_0.

E5 (discussion). Repeat E1 with an implicit solver such as BDF. Why is the result less sensitive to the tolerance, and what does it cost?

8 Conclusions

Fitting an ODE model to data becomes reliable with five habits: estimate log-parameters and scale the residuals to the noise; set the relative solver tolerance to 10^{-6}; start from gradient matching or from about ten spread initial guesses; look at the residuals before believing the fit; and report intervals on the log scale, checked by a profile for the weak parameters. Of these, the solver tolerance is the easiest to overlook, because the default of 10^{-3} gives no error message, and on an oscillating model it roughly halves the success rate of the default explicit solver. Together the five steps took every one of 400 test fits to the best solution and turned two classic data sets into estimates with honest intervals, together with a clear signal of where each model can be improved.

Checklist

☐ Parameters estimated on the log scale; residuals scaled to the noise.
☐ Relative tolerance 10^{-6}; estimate unchanged when the tolerance is reduced tenfold.
☐ Start from gradient matching (all states observed) or from about ten spread guesses.
☐ Residual plot shows no pattern; lag-1 autocorrelation below $2/\sqrt{n}$.
☐ Intervals on the log scale; profile computed for parameters with wide intervals.

Statements and Declarations

Funding. No funding was received for conducting this study.

Competing interests. The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The influenza data are the series digitised from the original report by A. A. King and co-workers, available in the repository `kingaa/sbied` (file `stochsim/bsflu_data.txt`). The lynx–hare data are available with the Stan case study [33]. Simulated data are generated with fixed random seeds by the archived scripts (Code availability).

Materials availability. Not applicable.

Code availability. Scripts and results that reproduce every figure, table and number are archived at Zenodo, https://doi.org/10.5281/zenodo.23501889, under the MIT licence.

Author contribution. Mohammad Jamhuri: conceptualisation, methodology, software, formal analysis, writing – original draft, supervision. Putri Wulan Agustina: investigation, validation, data curation, visualisation, writing – review and editing. Both authors read and approved the final manuscript.

Use of a large language model. An AI assistant was used in preparing this work; its use is documented in Appendix A. The authors take full responsibility for the content.

Appendix A Details of the numerical experiments

Use of an AI assistant.

The authors used an AI assistant (Claude, Anthropic, 2026) under their direction to write and run the simulation and analysis scripts, to prepare the figures, to search and verify the literature, and to draft parts of the text. The authors chose the topic, the test problems and the data sets, checked every numerical result by rerunning the archived scripts, verified each reference against its source, and revised the full text. They take responsibility for the final version of the manuscript.

Optimiser and solver.

All fits used a trust-region least-squares method [36] with forward-difference Jacobians and an adaptive Runge–Kutta solver of order 5(4), as provided by SciPy 1.15 [14]. Each simulation was limited to 20 000 right-hand-side evaluations and counted as a failure beyond that limit. Absolute tolerances were 10^{-3} times the relative tolerance times the typical state size, except in S0 and S1, which used the default 10^{-6}.

Grid (Fig. 4).

Lotka–Volterra and SIR with 2% noise; 50 initial guesses drawn log-uniformly within a factor $e^{0.7}$ of the truth; tolerances $10^{-3}, 10^{-4}, \ldots, 10^{-8}, 10^{-10}$; relative difference steps default, $10^{-6}, \ldots, 10^{-2}$. Success: all parameters within 5% of the global optimum.

Seeds, solvers and optimisers (Table 2).

Lotka–Volterra with 2% noise; three data sets (random seeds 0, 1 and 2), each with its own 50 initial guesses drawn as for the grid; solvers RK45, LSODA and BDF; trust-region and Levenberg–Marquardt least squares; absolute tolerance 10^{-3} times the relative tolerance; at most 40 000 right-hand-side evaluations per simulation. The global optimum of each data set was obtained from the true parameters with relative tolerance 10^{-10}. Jacobian errors were measured as for Fig. 3 at the default difference step.

Ablation (Fig. 5).

100 trials per model; in each trial new data with 5% multiplicative noise and ten initial guesses drawn log-uniformly within a factor e of the truth; S0–S2 used the first guess, S3 all ten. The reference minimum of each trial was the smallest SSE over all its fits and a fit started at the truth. The naive fits (S0) were also judged against the minimum of their own unweighted objective; the success counts were identical.

Gradient matching.

Cubic smoothing splines with smoothing parameter $n\sigma^2$, where σ is the noise level on the scale of the data (Lotka–Volterra test model) or of their logarithms (lynx–hare); derivatives evaluated at 200–400 equally spaced points; parameters from linear least squares as in (3).

Influenza model comparison.

SIR models with $k = 1, \dots, 4$ bed-confined stages, each left at rate $k\gamma$, and SEIR variants with a latent rate σ; square-root residuals, log-parameters, relative tolerance 10^{-6} with the LSODA solver, and 20 initial guesses per model. AICc $= m \log(\mathrm{SSE}/m) + 2n_p + 2n_p(n_p + 1)/(m - n_p - 1)$ with $m = 14$.

Real data.

50 initial guesses drawn log-uniformly in boxes $R_0 \in [1.1, 20]$, $\gamma \in [0.05, 3]$, $I_0 \in [0.1, 20]$ (influenza) and $a, c \in [0.1, 2]$, $b, d \in [0.005, 0.1]$, $h_0 \in [10, 100]$, $l_0 \in [1, 20]$ (lynx–hare). Profiles were computed on grids of step 0.05 in $\log p_i$, and the interval limits were obtained by linear interpolation.

Appendix B Answers to the exercises

E1.

With an explicit Runge–Kutta solver, about one third of the fits succeed at 10^{-3} (15–18 of 50 in our three data sets) and about two thirds at 10^{-6} (29–34 of 50), with two to three times fewer model evaluations at the tighter tolerance (Table 2). The remaining failures start in another basin of the cost; Step 3 removes them.

E2.

The tolerance. At 10^{-3} the relative Jacobian error is about 30% for every step between 10^{-10} and 10^{-3}; at 10^{-6} it is about 4×10^{-5} (Fig. 3).

E3.

With a smoothing level matched to about 10% noise on the log scale, gradient matching gives $(a, b, c, d) \approx (0.507, 0.0252, 0.746, 0.0230)$, within 8% of the final estimate (0.540, 0.0272, 0.796, 0.0237). The full fit from this start converges in 9 model evaluations.

E4.

For the SIR model, the lag-1 autocorrelation is 0.66, above $2/\sqrt{14} = 0.53$, and $R_0 = 3.96$. The two-stage model gives 0.34, a lower AICc (3.3 against 10.3) and $R_0 = 3.18$ (2.82–3.65).

E5.

BDF computes the Jacobian more accurately at loose tolerance (3% error at 10^{-3} for this problem), so 28–33 of 50 fits succeed already at 10^{-3}; each fit takes three to five times longer. The rule of Step 2 works for every solver, so it is the safer habit.

References

[1] Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelke, M., Kaschek, D., Hug, S., Kreutz, C., Harms, B.D., Theis, F.J., Klingmüller, U., Timmer, J.: Lessons learned from quantitative dynamical modeling in systems biology. PLoS ONE **8**(9), 74335 (2013) https://doi.org/10.1371/journal.pone.0074335

[2] Villaverde, A.F., Pathirana, D., Fröhlich, F., Hasenauer, J., Banga, J.R.: A protocol for dynamic model calibration. Briefings in Bioinformatics **23**(1) (2021) https://doi.org/10.1093/bib/bbab387

[3] Villaverde, A.F., Fröhlich, F., Weindl, D., Hasenauer, J., Banga, J.R.: Benchmarking optimization methods for parameter estimation in large kinetic models. Bioinformatics **35**(5), 830–838 (2018) https://doi.org/10.1093/bioinformatics/bty736

[4] Hass, H., Loos, C., Raimúndez-Álvarez, E., Timmer, J., Hasenauer, J., Kreutz, C.: Benchmark problems for dynamic modeling of intracellular processes. Bioinformatics **35**(17), 3073–3082 (2019) https://doi.org/10.1093/bioinformatics/btz020

[5] Raue, A., Steiert, B., Schelker, M., Kreutz, C., Maiwald, T., Hass, H., Vanlier, J., Tönsing, C., Adlung, L., Engesser, R., Mader, W., Heinemann, T., Hasenauer, J., Schilling, M., Höfer, T., Klipp, E., Theis, F., Klingmüller, U., Schöberl, B., Timmer, J.: Data2dynamics: a modeling environment tailored to parameter estimation in dynamical systems. Bioinformatics **31**(21), 3558–3560 (2015) https://doi.org/10.1093/bioinformatics/btv405

[6] Schälte, Y., Fröhlich, F., Jost, P.J., Vanhoefer, J., Pathirana, D., Stapor, P., Lakrisenko, P., Wang, D., Raimúndez, E., Merkt, S., Schmiester, L., Städter, P., Grein, S., Dudkin, E., Doresic, D., Weindl, D., Hasenauer, J.: pypesto: a modular

and scalable tool for parameter estimation for dynamic models. Bioinformatics **39**(11) (2023) https://doi.org/10.1093/bioinformatics/btad711

[7] Earn, D.J.D., Park, S.W., Bolker, B.M.: Fitting epidemic models to data: A tutorial in memory of fred brauer. Bulletin of Mathematical Biology **86**(9) (2024) https://doi.org/10.1007/s11538-024-01326-9

[8] Chowell, G.: Fitting dynamic models to epidemic outbreaks with quantified uncertainty: A primer for parameter uncertainty, identifiability, and forecasts. Infectious Disease Modelling **2**(3), 379–398 (2017) https://doi.org/10.1016/j.idm.2017.08.001

[9] Simpson, M.J., Baker, R.E.: Parameter identifiability, parameter estimation, and model prediction for differential equation models. SIAM Review **68**(1), 153–171 (2026) https://doi.org/10.1137/24m1667968

[10] Simpson, M.J., Maclaren, O.J.: Profile-wise analysis: A profile likelihood-based workflow for identifiability analysis, estimation, and prediction with mechanistic mathematical models. PLOS Computational Biology **19**(9), 1011515 (2023) https://doi.org/10.1371/journal.pcbi.1011515

[11] Murphy, R.J., Maclaren, O.J., Simpson, M.J.: Implementing measurement error models with mechanistic mathematical models in a likelihood-based framework for estimation, identifiability analysis and prediction in the life sciences. Journal of The Royal Society Interface **21**(210) (2024) https://doi.org/10.1098/rsif.2023.0402

[12] Lettermann, L., Jurado, A., Betz, T., Wörgötter, F., Herzog, S.: Tutorial: a beginner's guide to building a representative model of dynamical systems using the adjoint method. Communications Physics **7**(1) (2024) https://doi.org/10.1038/s42005-024-01606-9

[13] O'Brien, T.E., Silcox, J.W.: Nonlinear regression modelling: A primer with applications and caveats. Bulletin of Mathematical Biology **86**(4), 40 (2024) https://doi.org/10.1007/s11538-024-01274-4

[14] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., Mulbregt, P., Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.-L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J.,

Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., Miranda Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y.: Scipy 1.0: fundamental algorithms for scientific computing in python. Nature Methods **17**(3), 261–272 (2020) https://doi.org/10.1038/s41592-019-0686-2

[15] Shampine, L.F., Reichelt, M.W.: The matlab ode suite. SIAM Journal on Scientific Computing **18**(1), 1–22 (1997) https://doi.org/10.1137/s1064827594276424

[16] Rackauckas, C., Nie, Q.: Differentialequations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. Journal of Open Research Software **5**(1), 15 (2017) https://doi.org/10.5334/jors.151

[17] Bates, D.M., Watts, D.G.: Nonlinear Regression Analysis and Its Applications. Wiley, New York (1988). https://doi.org/10.1002/9780470316757

[18] Bock, H.G.: Numerical treatment of inverse problems in chemical reaction kinetics. In: Ebert, K.H., Deuflhard, P., Jäger, W. (eds.) Modelling of Chemical Reaction Systems. Springer Series in Chemical Physics, vol. 18, pp. 102–125. Springer, Berlin, Heidelberg (1981). https://doi.org/10.1007/978-3-642-68220-9_8

[19] Soetaert, K., Petzoldt, T., Setzer, R.W.: Solving differential equations in R: Package ¡b¿desolve¡/b¿. Journal of Statistical Software **33**(9) (2010) https://doi.org/10.18637/jss.v033.i09

[20] Leis, J.R., Kramer, M.A.: The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations. ACM Transactions on Mathematical Software **14**(1), 45–60 (1988) https://doi.org/10.1145/42288.46156

[21] Varah, J.M.: A spline least squares method for numerical parameter estimation in differential equations. SIAM Journal on Scientific and Statistical Computing **3**(1), 28–46 (1982) https://doi.org/10.1137/0903003

[22] Ramsay, J.O., Hooker, G., Campbell, D., Cao, J.: Parameter estimation for differential equations: a generalized smoothing approach. Journal of the Royal Statistical Society Series B: Statistical Methodology **69**(5), 741–796 (2007) https://doi.org/10.1111/j.1467-9868.2007.00610.x

[23] Liang, H., Wu, H.: Parameter estimation for differential equation models using a framework of measurement error in regression models. Journal of the American Statistical Association **103**(484), 1570–1583 (2008) https://doi.org/10.1198/

016214508000000797

[24] Dattner, I.: Differential equations in data analysis. WIREs Computational Statistics **13**(6) (2020) https://doi.org/10.1002/wics.1534

[25] Bortz, D.M., Messenger, D.A., Dukic, V.: Direct estimation of parameters in ode models using wendy: Weak-form estimation of nonlinear dynamics. Bulletin of Mathematical Biology **85**(11) (2023) https://doi.org/10.1007/s11538-023-01208-6

[26] Heitzman-Breen, N., Dukic, V., Bortz, D.M.: A practical identifiability criterion leveraging weak-form parameter estimation. Bulletin of Mathematical Biology **88**(5) (2026) https://doi.org/10.1007/s11538-026-01639-x

[27] Venzon, D.J., Moolgavkar, S.H.: A method for computing profile-likelihood-based confidence intervals. Applied Statistics **37**(1), 87 (1988) https://doi.org/10.2307/2347496

[28] Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., Timmer, J.: Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. Bioinformatics **25**(15), 1923–1929 (2009) https://doi.org/10.1093/bioinformatics/btp358

[29] Kreutz, C., Raue, A., Kaschek, D., Timmer, J.: Profile likelihood in systems biology. The FEBS Journal **280**(11), 2564–2571 (2013) https://doi.org/10.1111/febs.12276

[30] Wilson, E.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association **22**(158), 209–212 (1927) https://doi.org/10.1080/01621459.1927.10502953

[31] Anonymous: Influenza in a boarding school. British Medical Journal **1**(6112), 587 (1978)

[32] Odum, E.P.: Fundamentals of Ecology. W. B. Saunders, Philadelphia (1953)

[33] Carpenter, B.: Predator–prey population dynamics: the Lotka–Volterra model in Stan. Stan case study (2018). https://mc-stan.org/learn-stan/case-studies/lotka-volterra-predator-prey.html

[34] Avilov, K.K., Li, Q., Lin, L., Demirhan, H., Stone, L., He, D.: The 1978 English boarding school influenza outbreak: where the classic SEIR model fails. Journal of The Royal Society Interface **21**(220), 20240394 (2024) https://doi.org/10.1098/rsif.2024.0394

[35] Linden, N.J., Kramer, B., Rangamani, P.: Bayesian parameter estimation for dynamical models in systems biology. PLOS Computational Biology **18**(10),

1010651 (2022) https://doi.org/10.1371/journal.pcbi.1010651

[36] Branch, M.A., Coleman, T.F., Li, Y.: A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. SIAM Journal on Scientific Computing **21**(1), 1–23 (1999) https://doi.org/10.1137/s1064827595289108