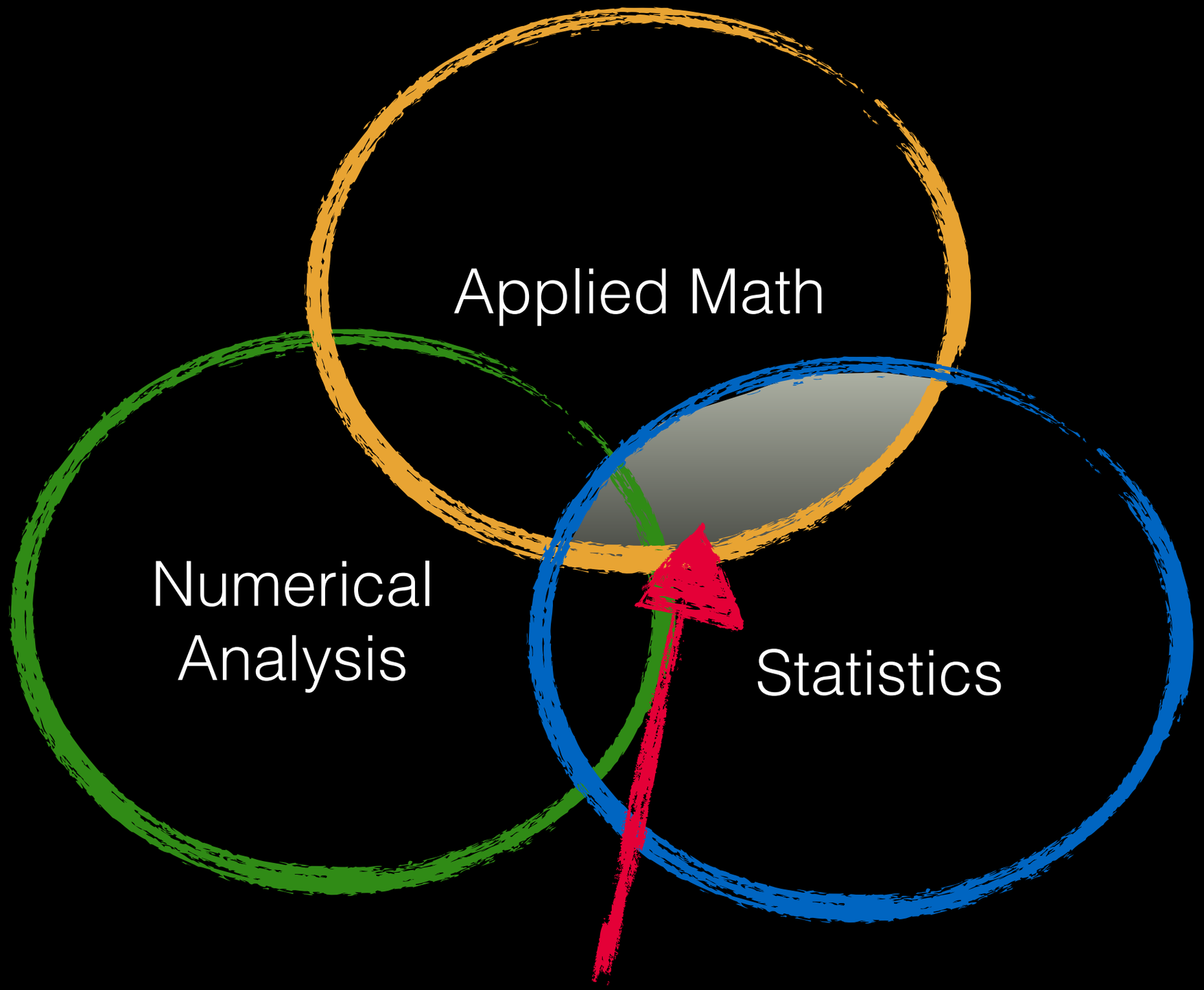
Probabilistically Solving Differential Equations (now with bonus stuff!!)

© Dave Campbell www.stat.sfu.ca/~dac5
dac5@sfu.ca

http://arxiv.org/abs/1306.2365



This is my happy research place

Parameter Estimation Methods; what to use and when to get fancy

- For model: $\dot{x}(t) = f[x(t), \theta]$
- With data: $y(t) = x(t) + \epsilon$, where: $\epsilon \sim N\left(0, \sigma^2\right)$
- We want to estimate parameter(s) θ

Model: $\dot{x}(t) = f[x(t), \theta]$ Data: $y(t) \sim N\left(x(t), \sigma^2\right)$

Nonlinear Regression

- As long as a Normal error structure is used nonlinear regression = nonlinear least squares

$$\hat{\theta} = arg\min_{\theta} \sum_{t} \left(y(t) - x(t, \theta)\right)^2$$

because this is minimizing the - log(likelihood)

point and variance estimates usually from ‘canned software’ in Matlab, R,…

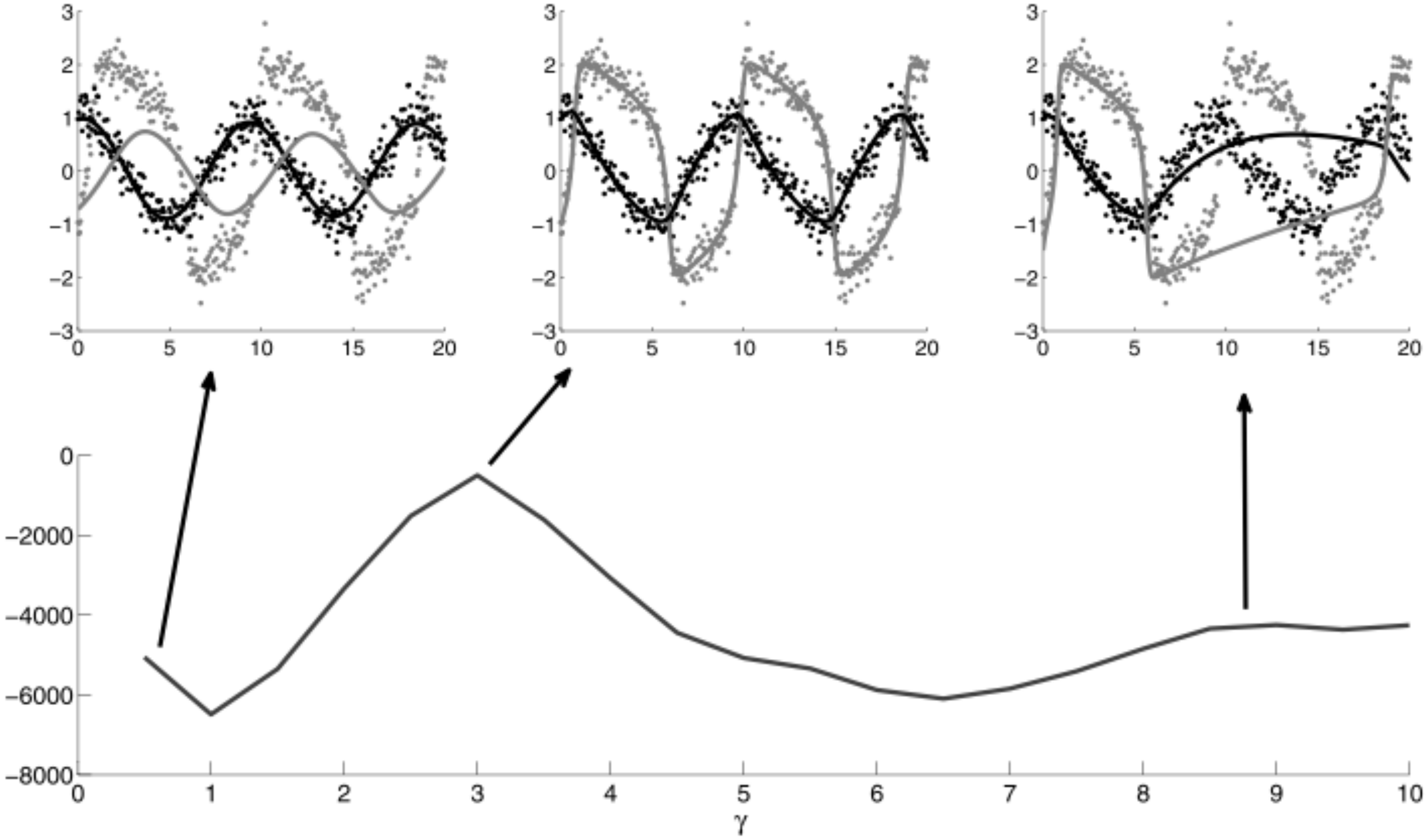
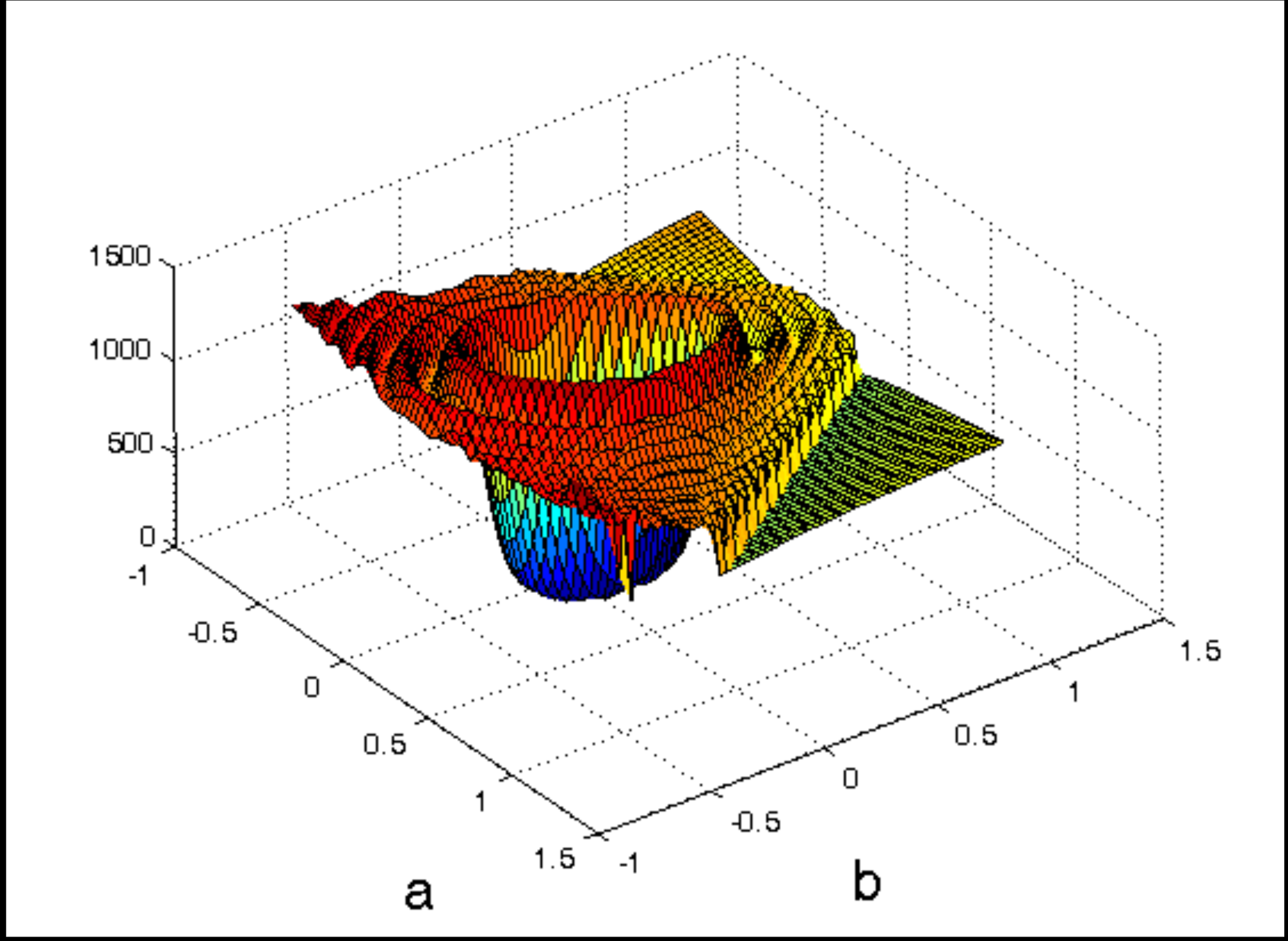


Fig. 1 A cross section of the FitzHugh-Nagumo log likelihood for γ (*bottom*) and the fits to the data for V (*grey*) and R (*black*) corresponding to the likelihood modes using the true parameter values (*top middle*), a small value (*top left*) and a large value (*top right*)

- Campbell, D., & Steele, R. J. (2011). Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, *22*, 429-443. doi:10.1007/s11222-011-9234-31111/j.1467-9868.2007.00610.x



- Ramsay, J. O., Hooker, G., Campbell, D., & Cao, J. (2007). Parameter estimation for differential equations:A generalized smoothing approach (with discussion). JRSS-B, 69(5), 741-796

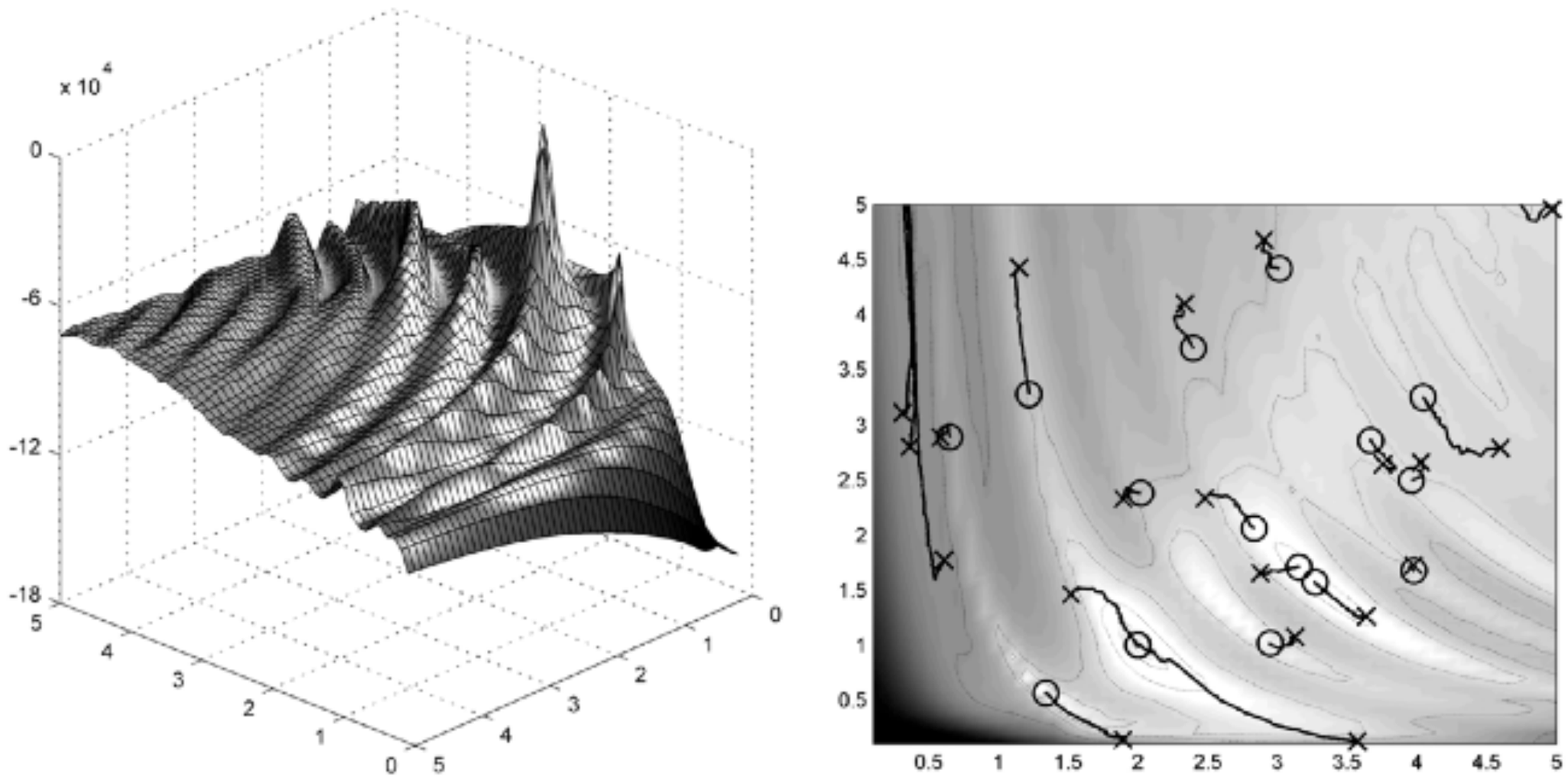
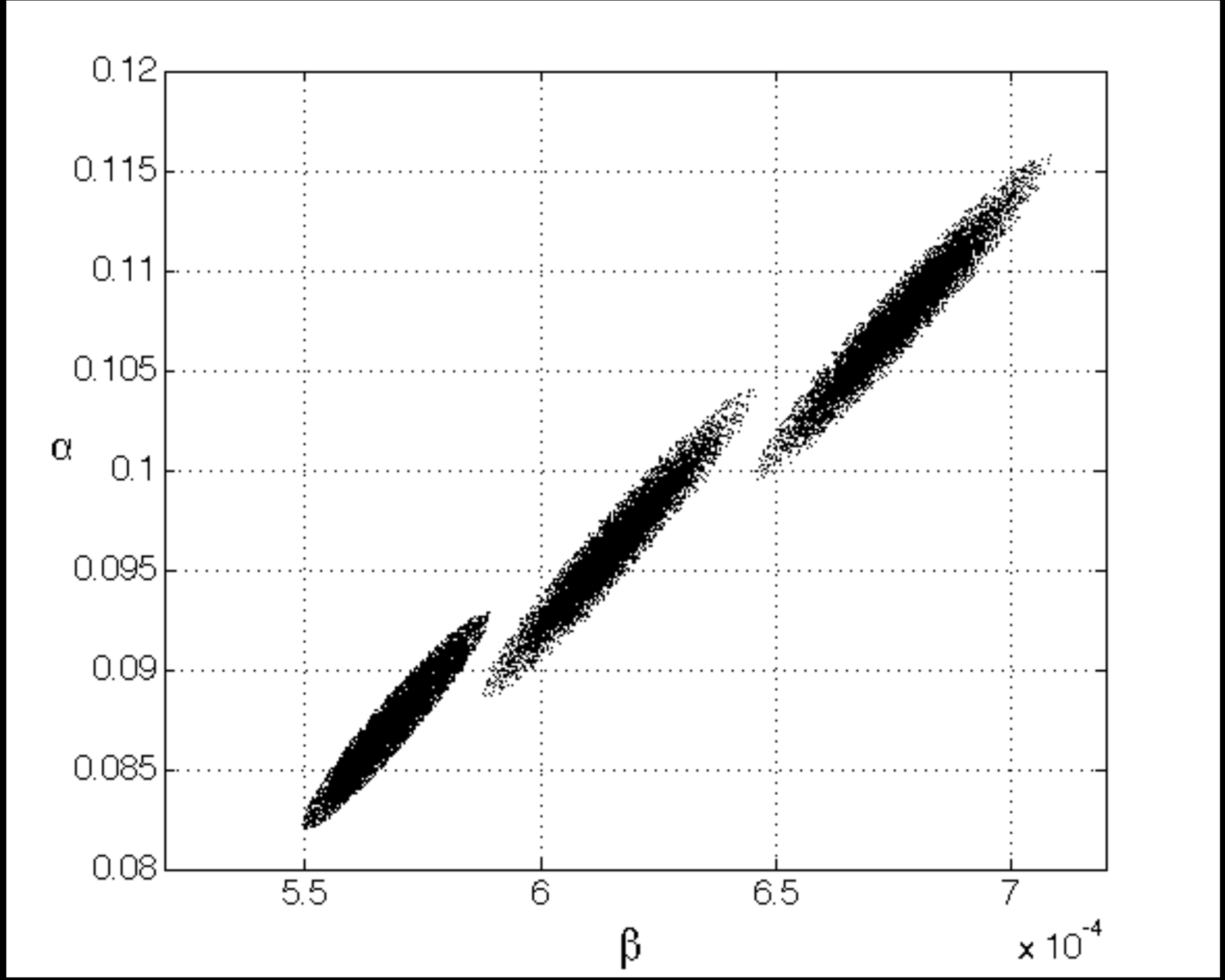


Fig. 4. Left plot: Posterior surface of two parameters of a Goodwin oscillator model (with z-axis in log-scale). Right plot: The progress of twenty independent Metropolis samplers, showing the starting positions (denoted by $\times$), path taken and finishing positions (denoted by $\circ$). The trapping of chains in local modes is most apparent.

- Calderhead, B., & Girolami, M. (2009). Estimating bayes factors via thermodynamic integration and population MCMC. Computational Statistics and Data Analysis, 53, 4028-4045.



- Campbell, D. A., & Lele, S. (2013). An ANOVA test for parameter estimability using data cloning with application to statistical inference for dynamic systems. Computational Statistics and Data Analysis. doi:10.1016/j.csda.2013.09.013

Nonlinear Regression

- plenty of 'canned software' in Matlab, R,…
- Can depend strongly on starting parameter values
- Unobserved states are fine
- Your model must be good - data cycles need to have fixed periods and amplitudes
- Variance estimates are good, confidence intervals generally use asymptotic assumptions that may not apply. (Fixable that using profile likelihood contours methods, but complicated - ask me later)
- Recommended strategy: use if you already kind of know parameters, your model is simple-ish, use this. Often try it 1st

Two Stage Methods

Model: $\dot{x}(t) = f[x(t), \theta]$ Data: $y(t) \sim N\left(x(t), \sigma^2\right)$

- Use a local linear regression method, start with:

$$arg \min_{\beta_0^{(t_0)}, \beta_1^{(t_0)}} \sum_t \underbrace{\exp\left[-\frac{1}{2K}(t - t_0)^2\right]}_{\text{weight function}} \underbrace{\left[y(t) - \beta_0^{(t_0)} + \beta_1^{(t_0)}(t - t_0)\right]^2}_{\text{linear regression}}$$

- Estimate $\hat{y}(t_0) = \beta_0^{(t_0)}$ and $\hat{\dot{y}}(t_0) = \beta_1^{(t_0)}$
- Then estimate $\hat{\theta}$ as $arg \min_\theta \sum_t \left(\hat{\dot{y}}(t) - f(\hat{y}(t), \theta)\right)^2$

Brunel, N. (2008). Parameter estimation of odes via nonparametric estimators. *Electronic Journal of Statistics*, *2*, 1242-1267.

Liang, H., & Wu, H. (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. Journal of the American Statistical Association, 103(484),

Two Stage Methods

- Use R, Matlab,… functions to smooth the data (keywords: lowess, loess, kernel smooth, local linear regression, smoothing,…), then optimization
- Fast but approximate (Discretization, and smoothing)
  - ok if your model is a little wrong
  - Must have all states measured
  - Variance estimates are bad, use bootstrap or another method
  - Recommended Strategy: Use 2-stage to get the neighbourhood of parameters, use them as a starting point in nonlinear regression to improve point and variance. Generally easy to get ballpark estimates

Model Relaxation Methods

- Use model based smoothing (trade off between solving the model and interpolating the data) through a basis expansion:

$$\tilde{x}(t) = \sum_i C_i \phi(t)$$

- Set up basis coefficients C_i as functions of θ and smoothing parameter λ:

$$arg \min_{C_1,\ldots,C_b} \underbrace{\sum_t \left(y(t) - \tilde{x}(t)\right)^2}_{\text{data fit}} + \underbrace{\lambda \int_T \left(\dot{\tilde{x}}(t) - f(\tilde{x}(t), \theta)\right)^2}_{\text{model fit}}$$

Generalized profiling approach

- Set up basis coefficients C_i as functions of a fixed θ and smoothing parameter λ:

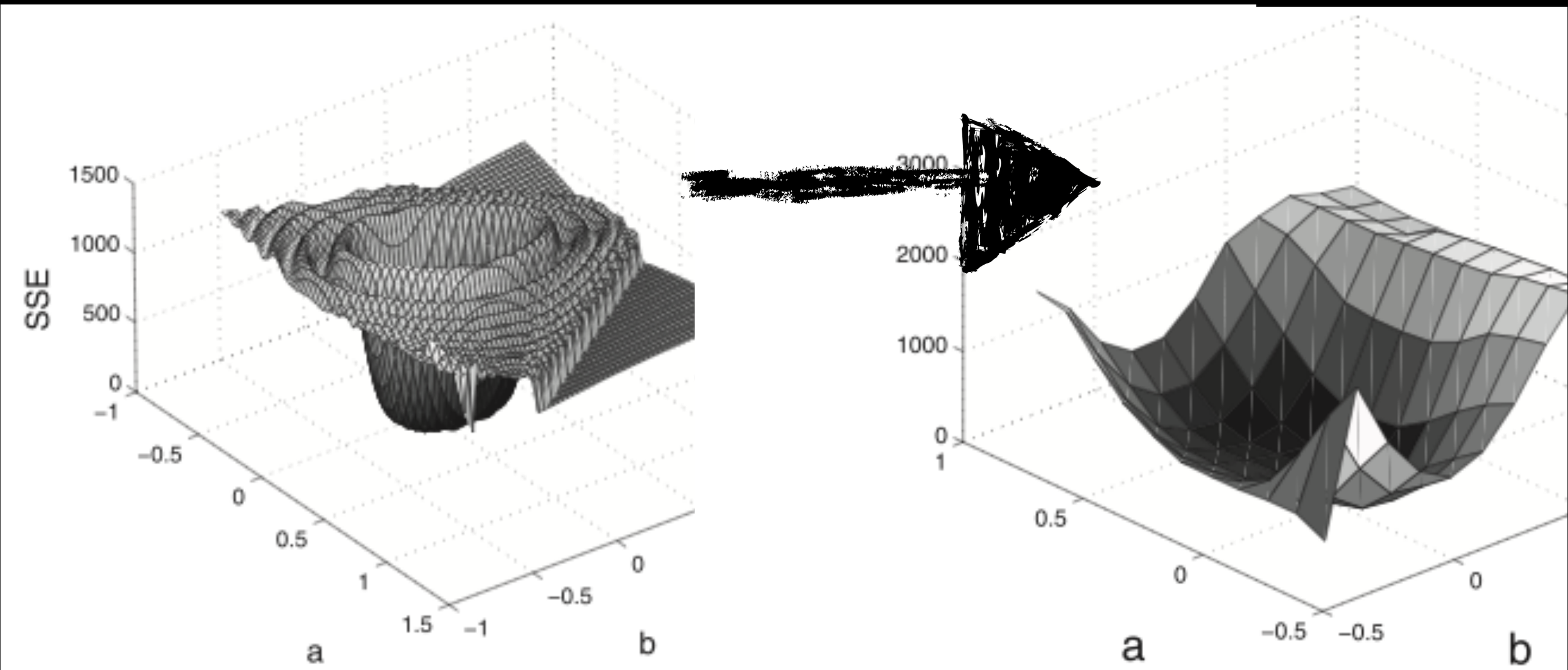
$$arg \min_{C_1,\ldots,C_b} \underbrace{\sum_t \left(y(t) - \tilde{x}(t)\right)^2}_{\text{data fit}} + \lambda \underbrace{\int_T \left(\dot{\tilde{x}}(t) - f(\tilde{x}(t), \theta)\right)^2}_{\text{model fit}}$$

- Fit parameters θ through:

$$arg \min_{\theta} \sum_t \left(y(t) - \tilde{x}(t)\right)^2$$

while maintaining C_is at their optima (profile over them)

Nonlinear regression vs generalized profiling



- Model relaxation towards the data makes estimation work.
- Must download software for Matlab, or use library(CollocInfer) in R (ask me I have working code to get it running) http://faculty.bscb.cornell.edu/~hooker/profile_webpages/
- Variance estimates from software can be bad if model is nonlinear (ask me)
- Ok if not all states measured
- Ok if model is wrong but useful, ok if cycles are close but not perfectly periodic
- Bayesian version available

- Suggested Use: If nonlinear least squares is doing strange things and you have unobserved states use this. Complexity suggests it's not a first try method. Software is good, but will take a few hours to set up.

Ramsay, J. O., Hooker, G., Campbell, D., & Cao, J. (2007). Parameter estimation for differential equations:A generalized smoothing approach (with discussion). Journal of the Royal Statistical Society Series B, 69(5), 741-796.

Campbell, D. A., & Chkrebtii, O. (2013). Maximum profile likelihood estimation of differential equation parameters through model based smoothing state estimates. *Mathematical Biosciences* doi:/10.1016/j.mbs.2013.03.011

Approximate Methods

- If your model is such that you can't calculate your likelihood (involves latent variables, SDE realizations…)

- Choose data summaries of interest '**S**' (number of spikes, average time between peaks, variance of function, autocorrelation…)

- Estimate θ based on $arg\min_{\theta} \sum_{s} (S(y) - S(x, \theta))^2$

Approximate Methods

Great papers on this:
Wood, S. (2010). *Statistical inference for noisy nonlinear ecological dynamic systems.* Nature, 466, 1102-1104. doi:10.1038/nature09319

Ryder, R., Robert, C. P., Pudlo, P., & Marin, J. M. (2011). *Approximate bayesian computational methods*. Statistics and Computing.

Example using complicated invasive species model:
O. Chkrebtii, E. Cameron, D. Campbell, E. Bayne “Transdimensional Approximate Bayesian Computation for Inference on Invasive Species Models with Latent Variables of Unknown Dimension” arXiv:1310.2888 (submitted)

Approximate Bayesian Computation & Synthetic Likelihood Methods

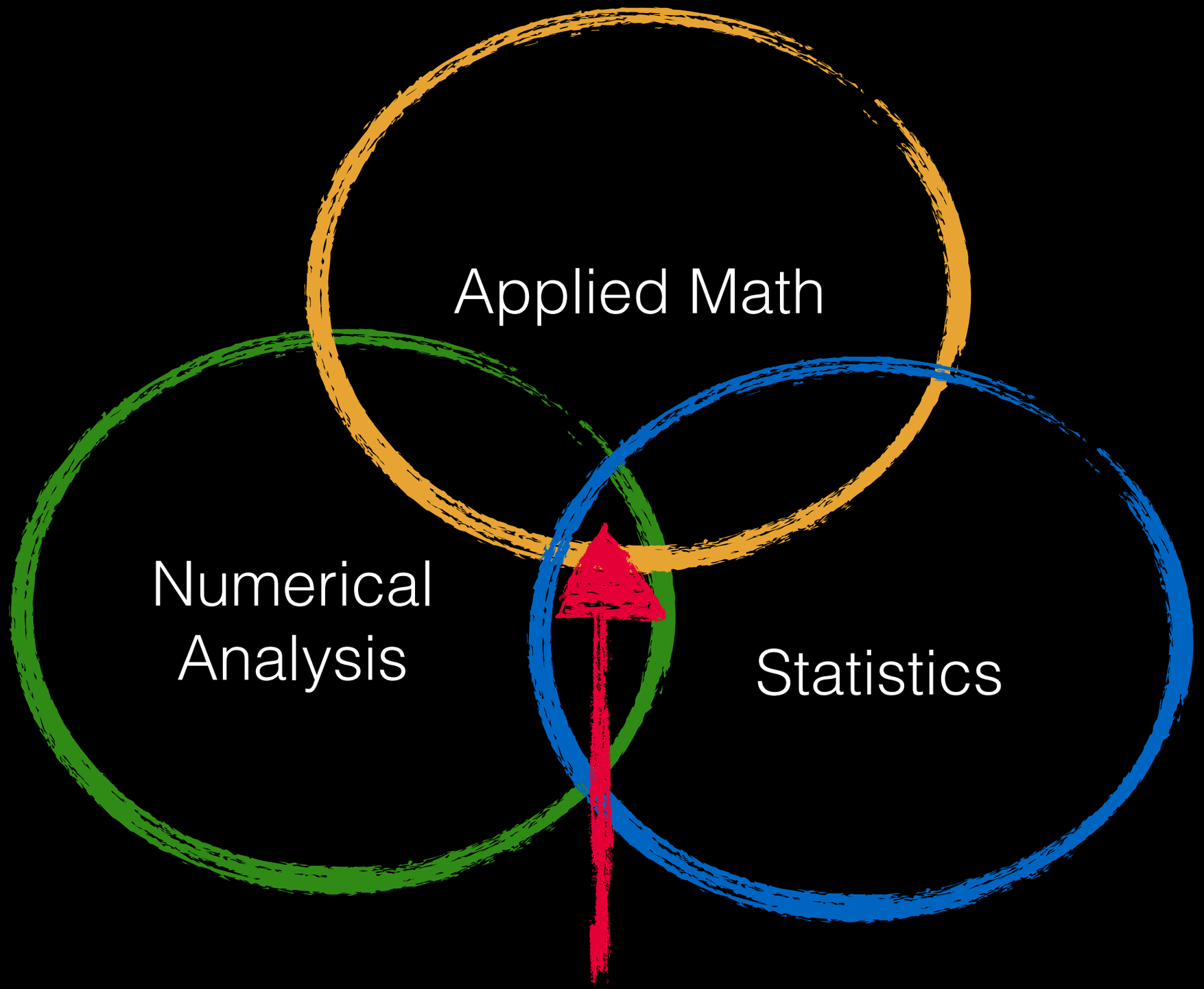
- Comes in Frequentist and Bayesian flavours
- Must be able to forward simulate the model (make fake data)
- Extracts useful information when the right thing to do is not possible
- Doesn’t play nice with model selection
- May lead to unidentifiable parameters
- No ‘canned software’, but quick to code for an expert
- Generally requires MCMC to get going (ask me)

- You always need to predict values from your model, solve your system or otherwise approximate a solution

- When the model is chaotic this step is toughest

Probabilistically Solving Differential Equation Models

http://arxiv.org/abs/1306.2365



This work is here

http://arxiv.org/abs/1306.2365

Outline

Motivation

- Things ‘chaotic people’ already noticed about chaotic systems

Probabilistic Solution for DE models

- Gaussian process: distributions on function spaces
- Bayesian statistics (prior, likelihood, posterior) as a solver

Loose Ends

- ode, pde, dde, and mixed boundary value problem
- To do list

Differential Equation problems

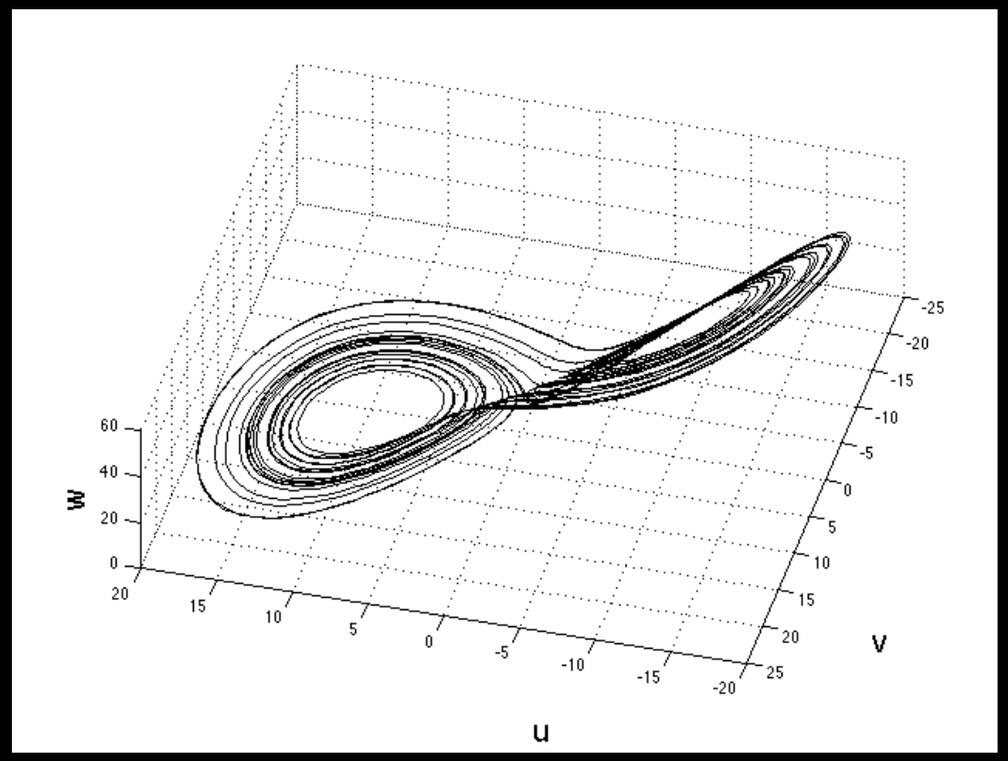
- Forward problem: predicting the model behaviour
- Inverse problem: Inferring the model parameters from data

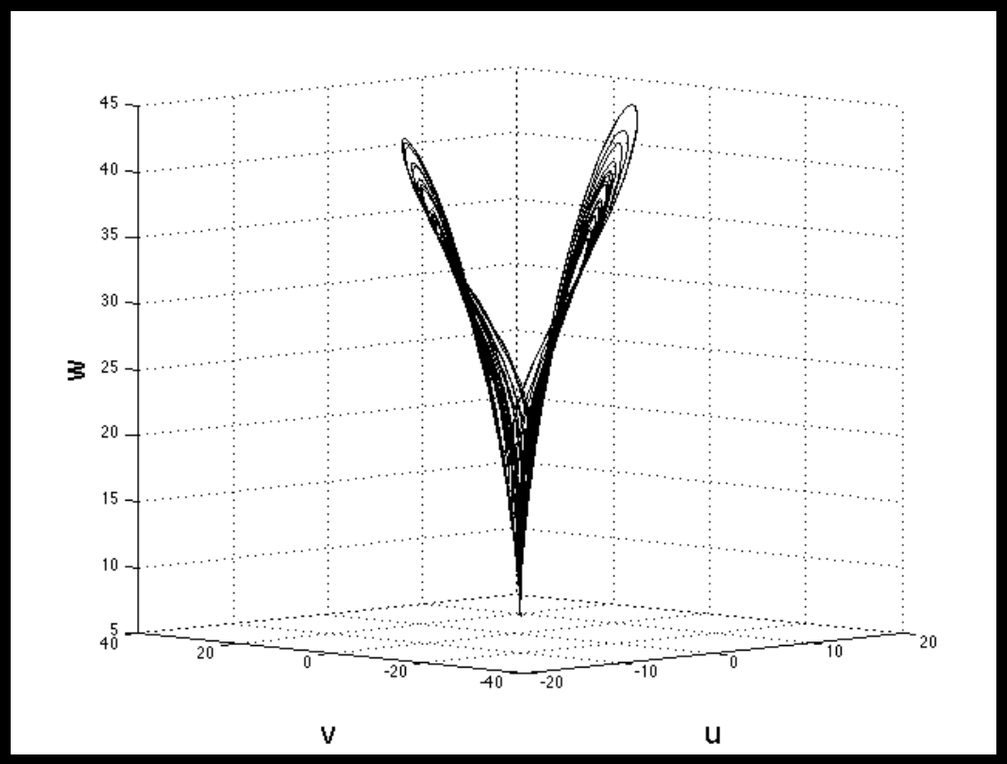
Lorenz System

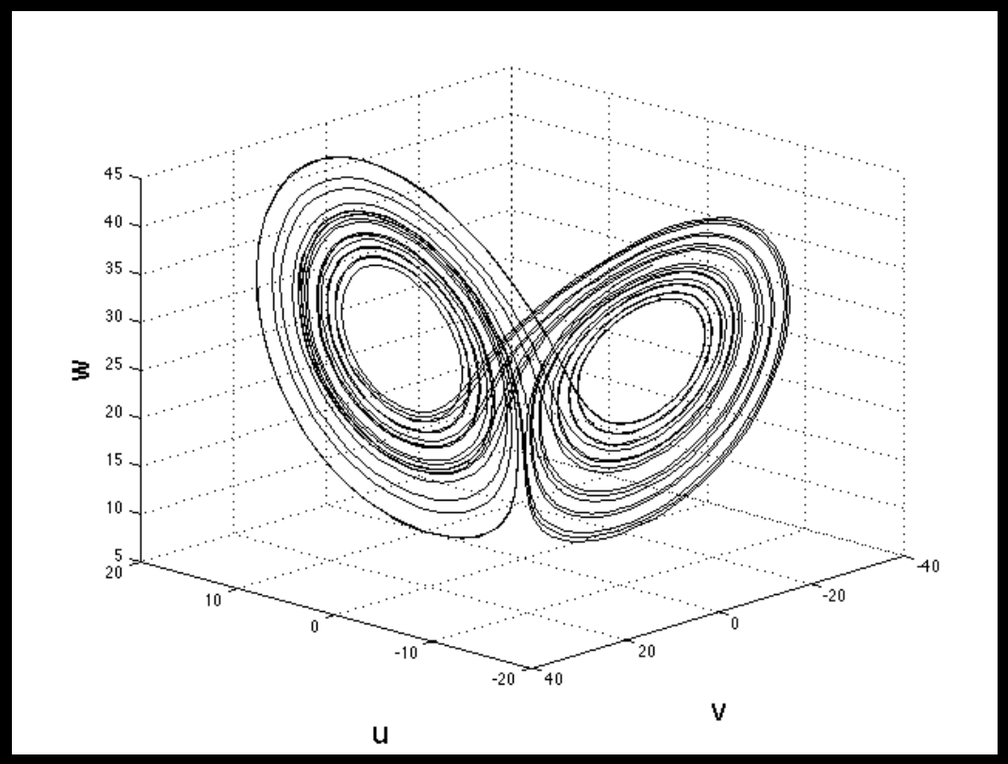
$$\frac{du}{dt} = -\theta u + \theta v$$

$$\frac{dv}{dt} = -\rho u - v - uw$$

$$\frac{dw}{dt} = uv - \beta w$$







Numerical solvers

- For model: $\frac{dx(t)}{dt} = f(x[t])$
- Set a discretization grid t=0,1,2,…T
- From start point $x(t_0)$ project ahead one step:

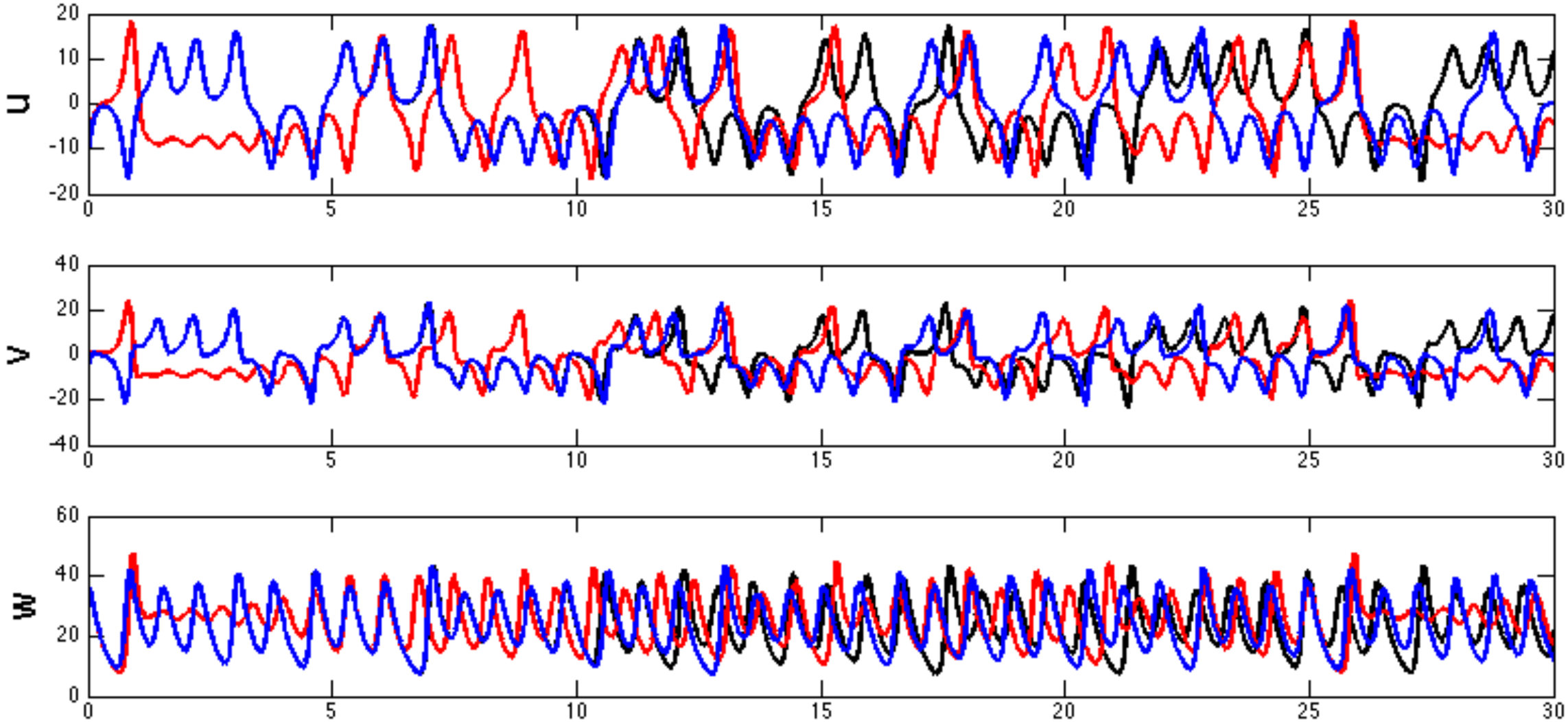
$$x(t_1) = x(t_0) + \Delta t * f(x[t_0])$$

- variants: take a few interim steps (i.e. trapezoid rule), use weighted projection based on past few points, (i.e. use higher order taylor expansion),…

Using Different solvers

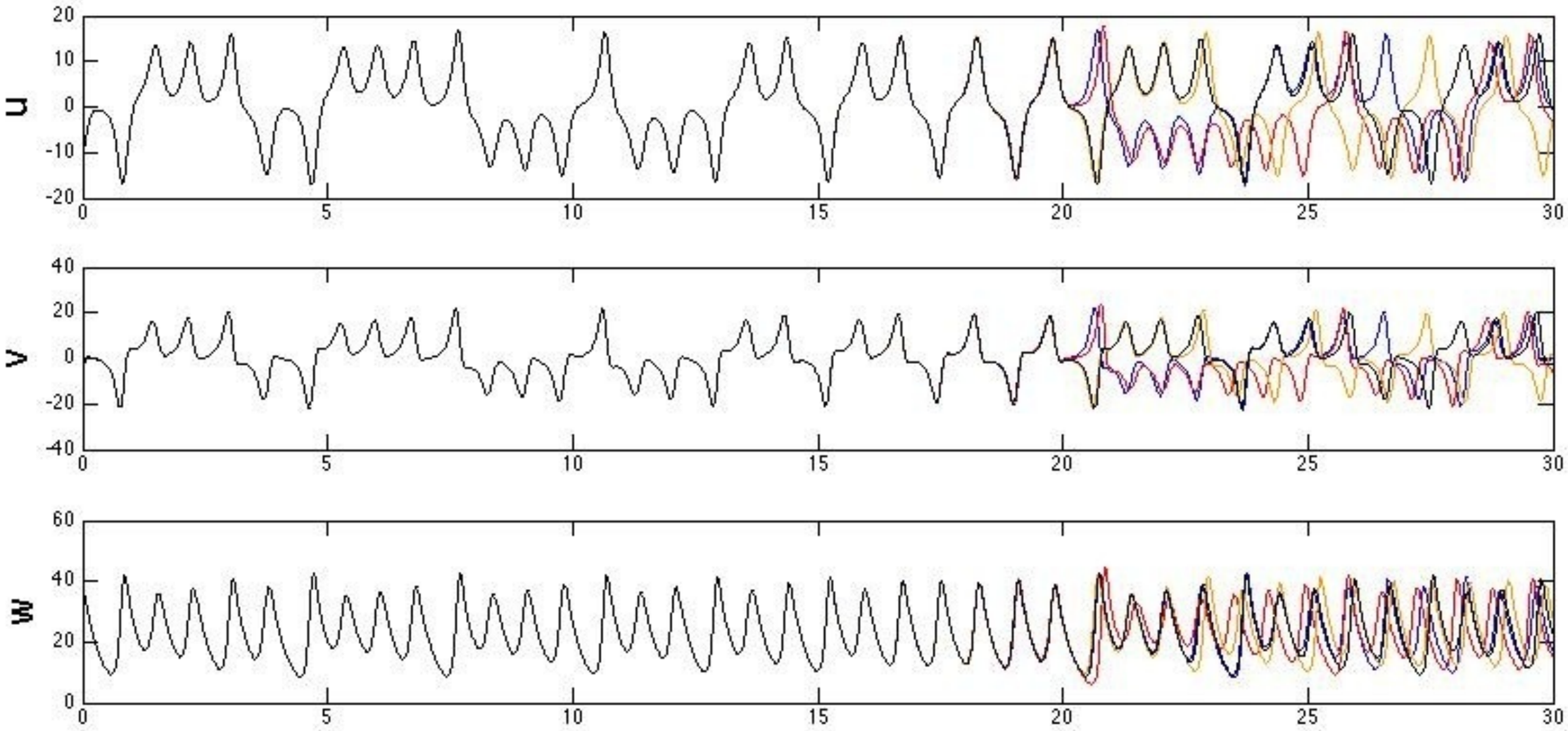
red: Euler, **black:Runge-Kutta with 4 interim steps,**

blue:Adams-Bashforth-Moulton PECE multistep solver using ‘several’ previous points

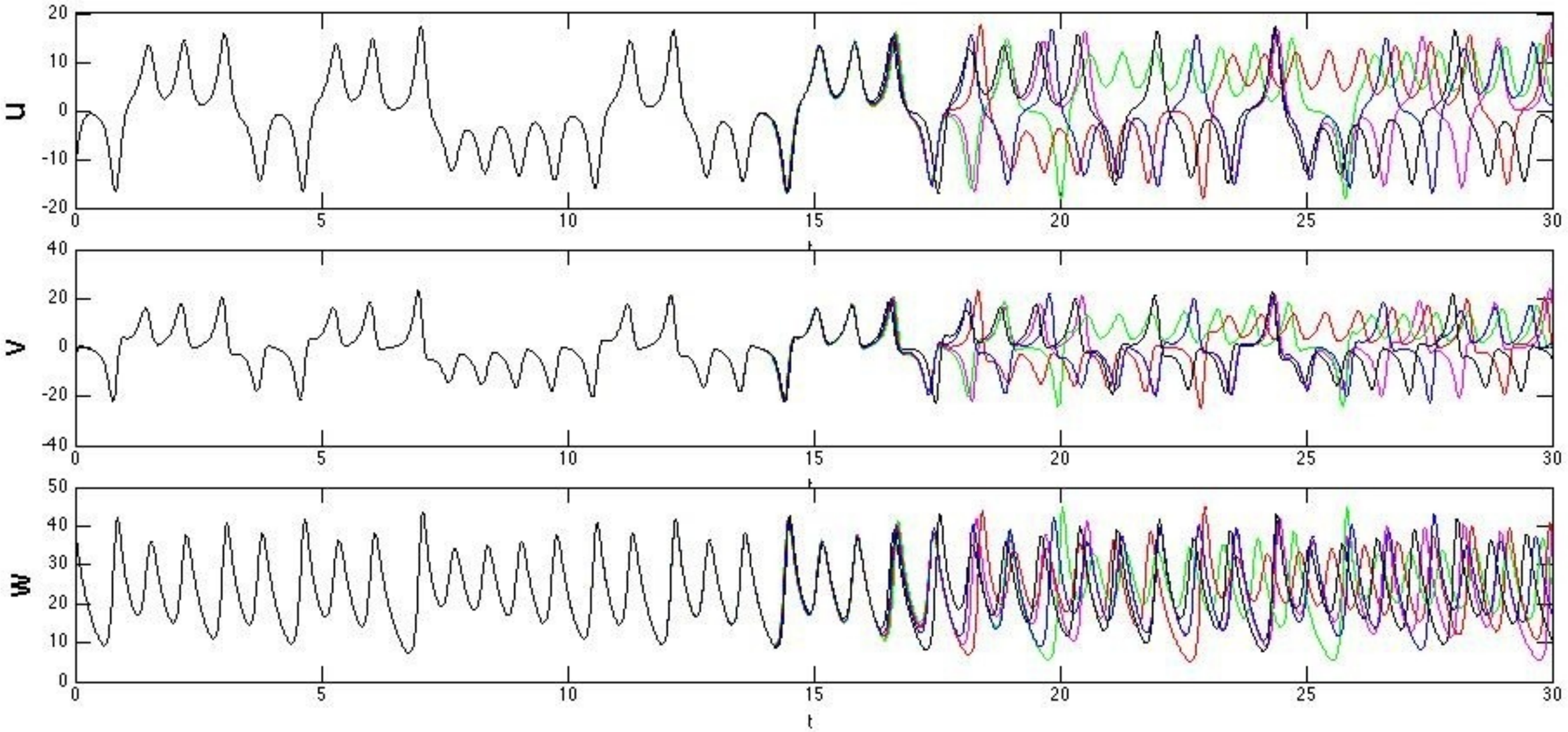


Solver Error propagation

Impact of adding 10^{-8} to the initial state of one component at a time



Fixed solver (4th order Runge Kutta), fixed initial condition, but changing error tolerance



”

From the point of view of inverse problems, the convergence of the forward model alone is not necessarily sufficient... As [grid step] h → 0+, the dimensionality of the approximation x(h) usually increases, i.e. n →∞. This means that the complexity of the inverse problem of estimating x(h) increases as the approximation improves. Hence, **when the forward model is accurate, the inverse problem may be prohibitively large to be tackled by any computational scheme. On the other hand, if the forward model is inaccurate, the discretization error may become significant compared to the measurement error. Together with the fact that the inverse problem is ill posed, the approximation error may destroy the quality of the estimate** of x(h).”

Arridge et al (2006). “Approximation errors and model reduction with an application in optical diffusion tomography”. Inverse Problems

One solver, fixed Error Tolerances

- There is a distribution of possible functions for a fixed amount of possible numerical error accumulation.

Implications of the chaos problem

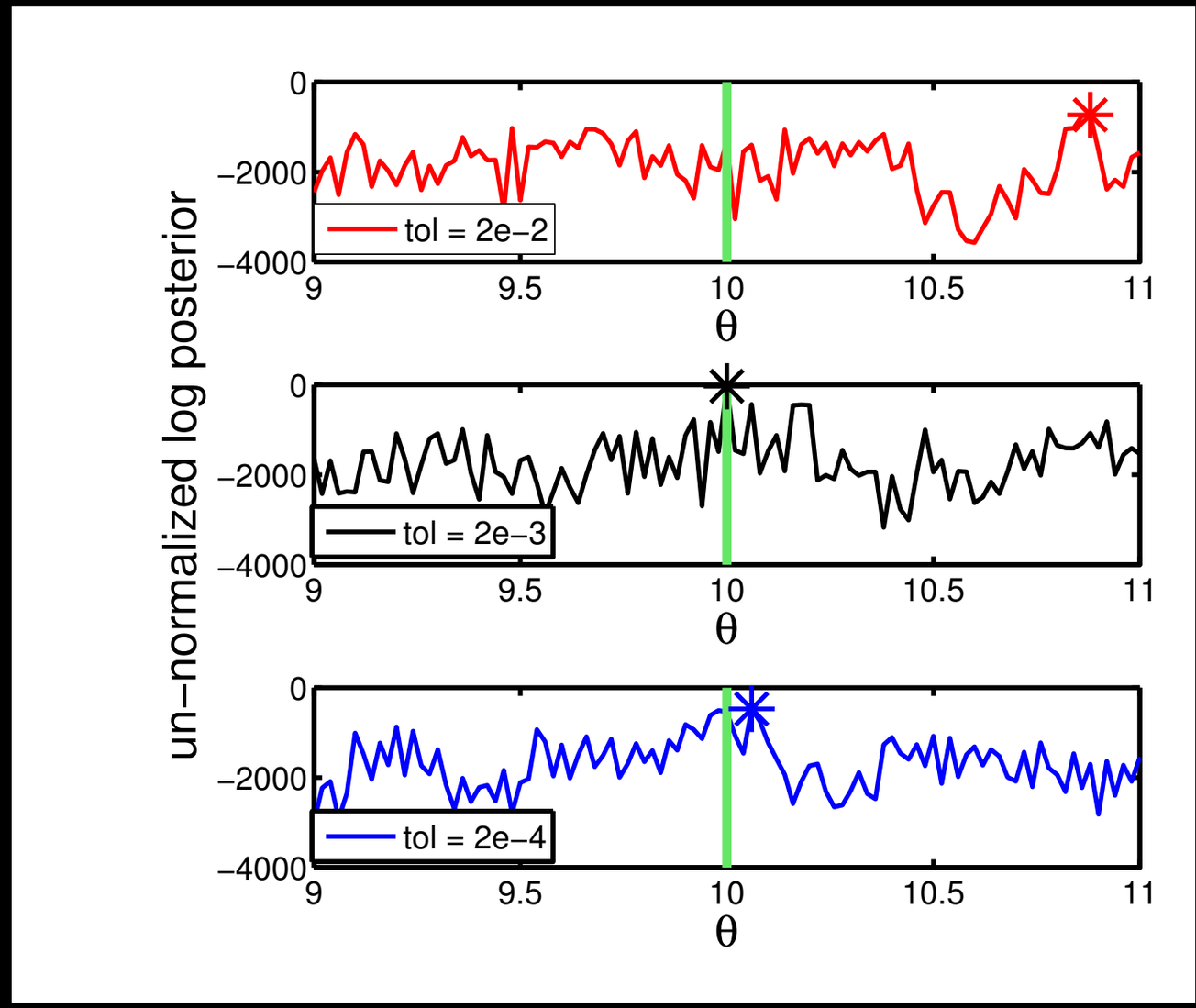
We use numerical solutions to infer parameter values, behaviours,… but for chaotic systems:.

- small errors propagate into divergent solutions
- changes in solver give different solutions
- changes in error tolerance either lead to infinitely long compute times or error prone quality solutions

But what does this mean for inference such as parameter estimates?

Impact on Parameter Estimation

- Single parameter Lorenz estimation problem



What does the computed numerical solution to a chaotic system tell us about long term behaviour?

Probabilistic Solution for DE models

- Goal: To solve a system of differential equations and quantify the uncertainty of that solution
- Method: Use a distribution on a function space and a sequential updating scheme to estimate a solution
- Tools: Gaussian Processes, Bayesian Methods

Bayesian Inference

- Step 1: define what you know about the solution to the problem a-priori (prior)
- Step 2: gather some new information (likelihood)
- Step 3: combine old and new information to get a posterior (posterior)

Prior
(before we have vector field evaluations)

Gaussian Process (GP) prior is a distribution for a function x(t) defined by a correlation:

Corr(x(t),x(s)) —>1 as ||s-t||—>0

Gaussian process prior:

- puts a distribution on possible solutions (restriction to a function space)
- like having a functional assumptions such as infinite differentiability, discontinuous derivatives at some points…

Prior
(before we have vector field evaluations)

Gaussian Process (GP) prior is a distribution for a function x(t) defined by a correlation:

Corr(x(t),x(s)) —>1 as ||s-t||—>0

- Use prior: $\dot{x}(s) \sim N(\mu_0^f, C_0^f)$ ≠ Brownian motion

Where the eigen-functions of the prior covariance C_0^f form a basis for the space containing the derivative of the solution

Prior

(before we have vector field evaluations)

Gaussian Process (GP) prior is a distribution for a function x(t) defined by a correlation:

Corr(x(t),x(s)) —>1 as ||s-t||—>0

- Use the prior: $\dot{x}(s) \sim N(\mu_0^f, C_0^f)$

Where the eigen-functions of the prior covariance C_0^f form a basis for the space containing the derivative of the solution

This defines a prior measure on the solution:

$$x(s) \sim N(\mu_0, C_0)$$

‘Data’ model (Likelihood)

goal is to solve a differential equation model

- grid (evaluation) points: $s_1, \ldots, s_N$
- ‘data’: approximate solutions points: $\tilde{x}(s_1), \ldots, \tilde{x}(s_N)$
- ‘data’: vector field evaluations at those grid points

$$f_{1:N} = [f(s_1, \tilde{x}(s_1)), \ldots, f(s_N, \tilde{x}(s_N))]$$

- Likelihood models the true derivative considering the ‘data’ known so far and the model vector field evaluation

$$Likelihood\,(\dot{x}(s)) \propto exp\left(-\frac{1}{2}||\dot{x}(s) - f_{1:N}||^2_{\Delta_N}\right)$$

Posterior for solution

$$p\left(x(t) \mid f_{1:N}, x_0, \Psi, \theta\right) \sim N\left(\mu_N(t), C_N\right)$$

- prior on solution x(t): a distribution on a space of functions
- Likelihood is based on ‘data’
- data: a set vector field evaluations (at points to be selected in later slides)
- Posterior: an updated distribution of x(t) given what we now know about the vector field $\boldsymbol{f}$, the model parameters $\boldsymbol{\theta}$, and the (updated) Gaussian Process parameters $\boldsymbol{\psi}$

Posterior for solution

$$p\left(x(t) \mid f_{1:N}, x_0, \Psi, \theta\right) \sim N\left(\mu_N(t), C_N\right)$$

- Combines the Gaussian Process prior with vector field evaluations (likelihood) to obtain a distribution of solutions (posterior)

- The posterior is also Gaussian Process with mean:

relating **x(t)** to the evaluation points **s**

and covariance:

between times **t** and **s** and involving derivatives and states

Loose ends

- How to get useful vector field evaluations
- Some chaotic examples
- Parameter Estimation

Jump ahead?

Vector field evaluations

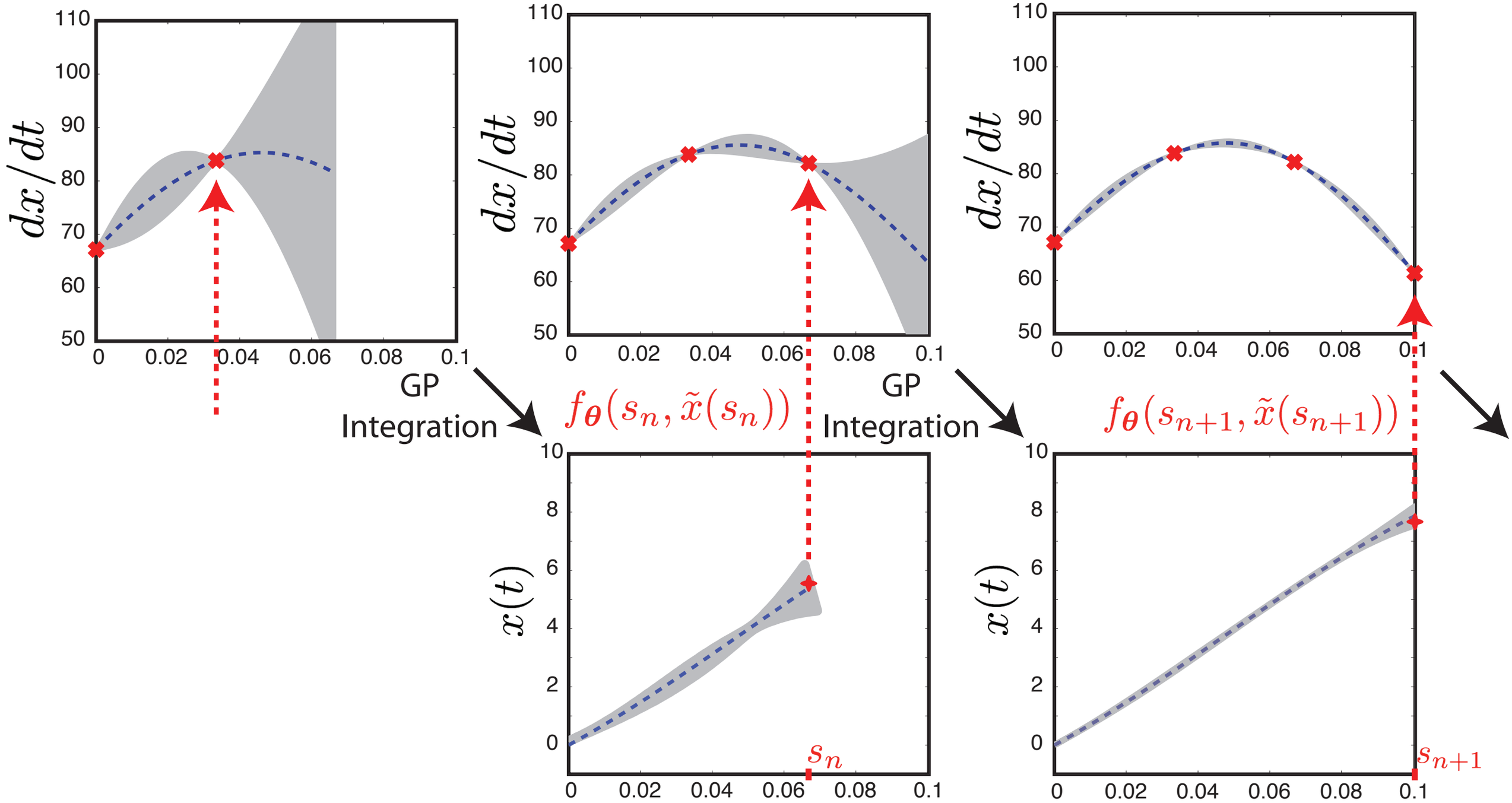
- The posterior distribution of the DE solution depends on vector field evaluation ‘data’ ***f***
- Completely random smattering of points 8^(
- Targeted points near useful locations 8^)

Probabilistic Solution for DE models

- Based on fixed boundary condition we get information about state $x(0)$ and derivative $\dot{x}(0)$

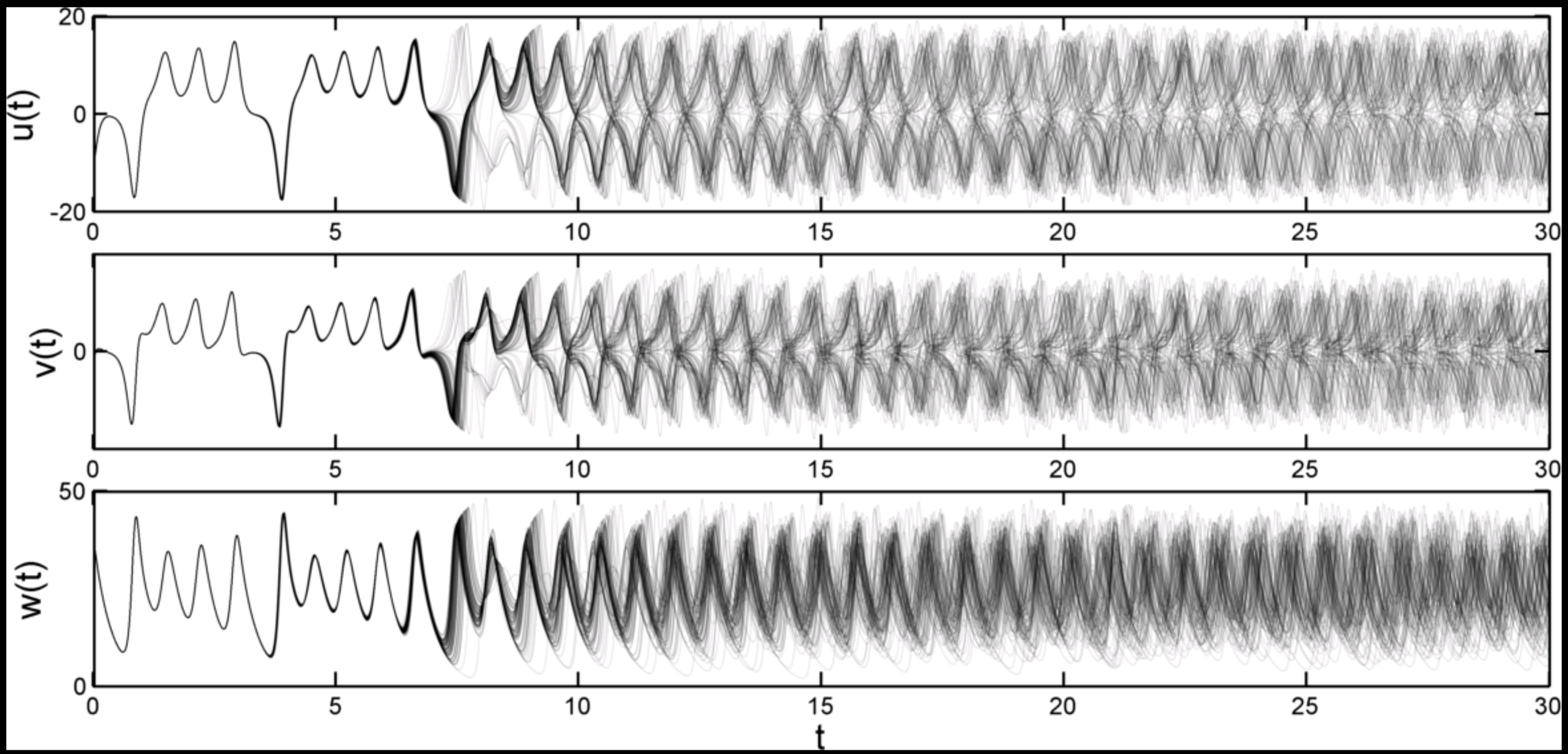
- We use the posterior to sample ahead to the next grid point $x(s)$ (using the posterior predictive distribution) and we compute its derivative $\dot{x}(s)$

- That point is our new data and we use that to update our information about the solution $x(t)$


dx/dt
110
100
90
80
70
60
50
0
0.02
0.04
0.06
0.08
0.1
GP
Integration
$f_{\boldsymbol{\theta}}(s_n, \tilde{x}(s_n))$
$x(t)$
10
8
6
4
2
0
s_n
GP
Integration
$f_{\boldsymbol{\theta}}(s_{n+1}, \tilde{x}(s_{n+1}))$
s_{n+1}

Lorenz System

Result is a distribution on the function space of solutions possible with a fixed initial state and solution grid



Parameter Estimation

For solution ***x(θ,t)*** we plug its distribution into the usual parameter estimation scheme. Giving the posterior:

$$p\big(\boldsymbol{\theta}, \mathbf{x}_0, \mathbf{x}(\mathbf{t}, \boldsymbol{\theta}), \mathbf{f}, \Psi | \mathbf{y}(\mathbf{t})\big) \propto \underbrace{p\big(\mathbf{y}(\mathbf{t}) | \mathbf{x}(\mathbf{t}, \boldsymbol{\theta})\big)}_{\text{Likelihood}} \times \underbrace{p\big(\mathbf{x}(\mathbf{t}, \boldsymbol{\theta}), \mathbf{f} | \boldsymbol{\theta}, \mathbf{x}_0, \Psi\big)}_{\text{Probabilistic Solution}} \times \underbrace{\pi\big(\boldsymbol{\theta}, \mathbf{x}_0, \Psi\big)}_{\text{Prior}}$$

Parameter Estimation

- For solution ***x(θ,t)*** we plug this into a likelihood to compare the solution to observations:

$$Likelihood(\theta \mid y) \propto \exp\left(-\frac{1}{2}\Sigma(y_t - x(\theta, t))^2\right)$$

- Include any prior information on the parameters:

$$\pi(\theta)$$

- Combine to get a Posterior Distribution for the Parameters

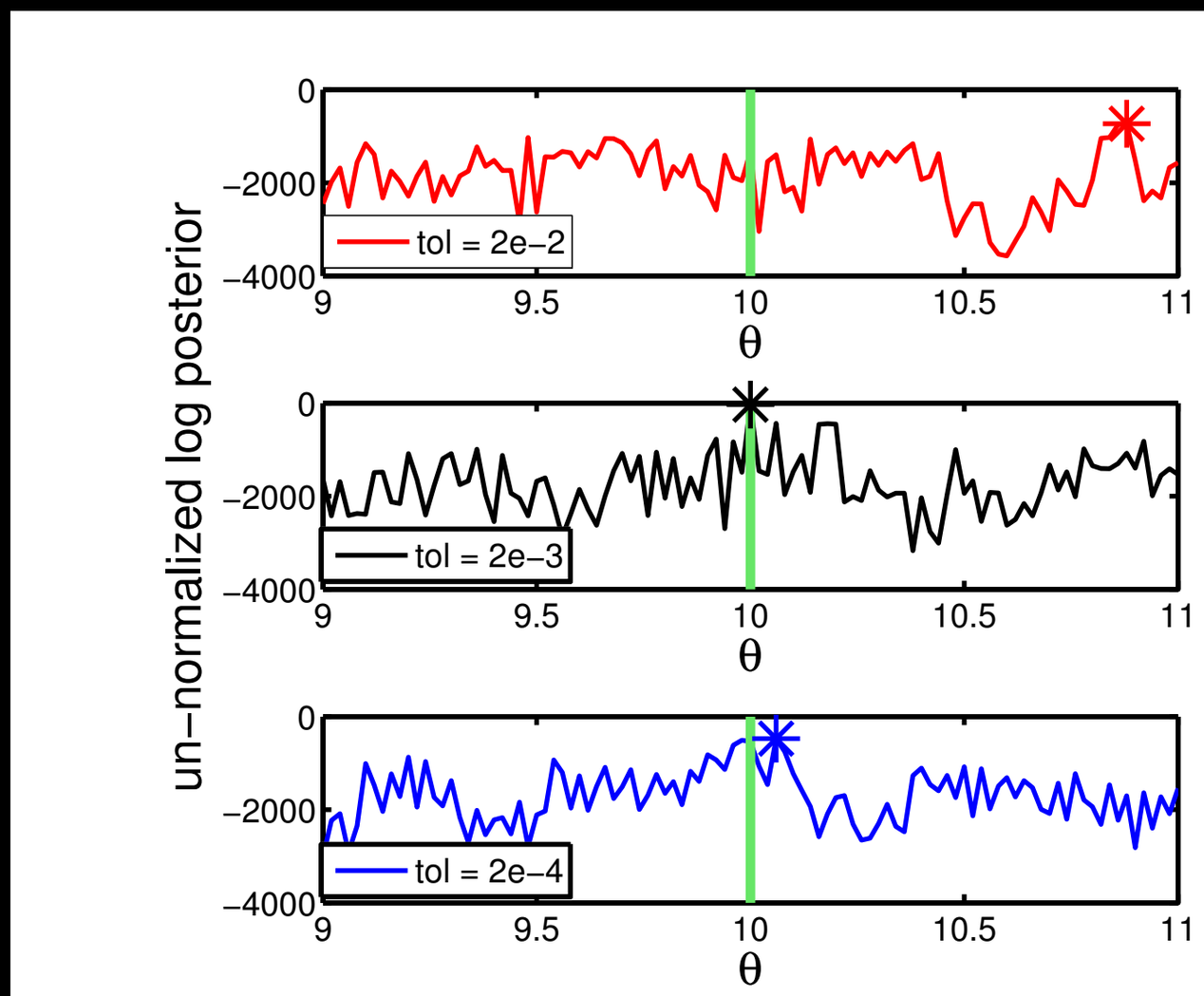
Parameter Estimation

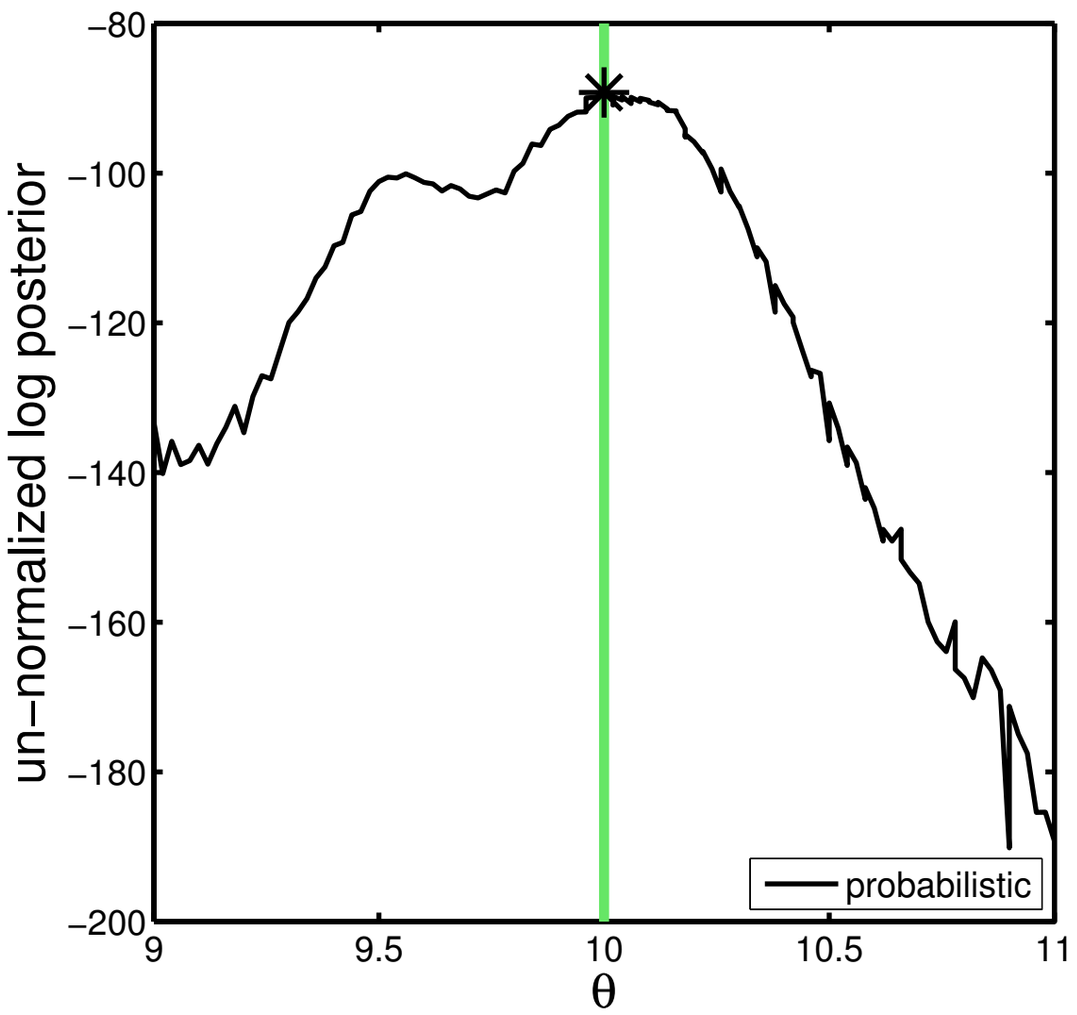
- Note that parameter estimation becomes a hierarchical process now that ***x(θ,t)*** has a distribution instead of a single (wrong?) value

$$Likelihood(\theta \mid y) \propto \exp\left(-\frac{1}{2}\Sigma(y_t - x(\theta, t))^2\right)$$

Impact on Parameter Estimation

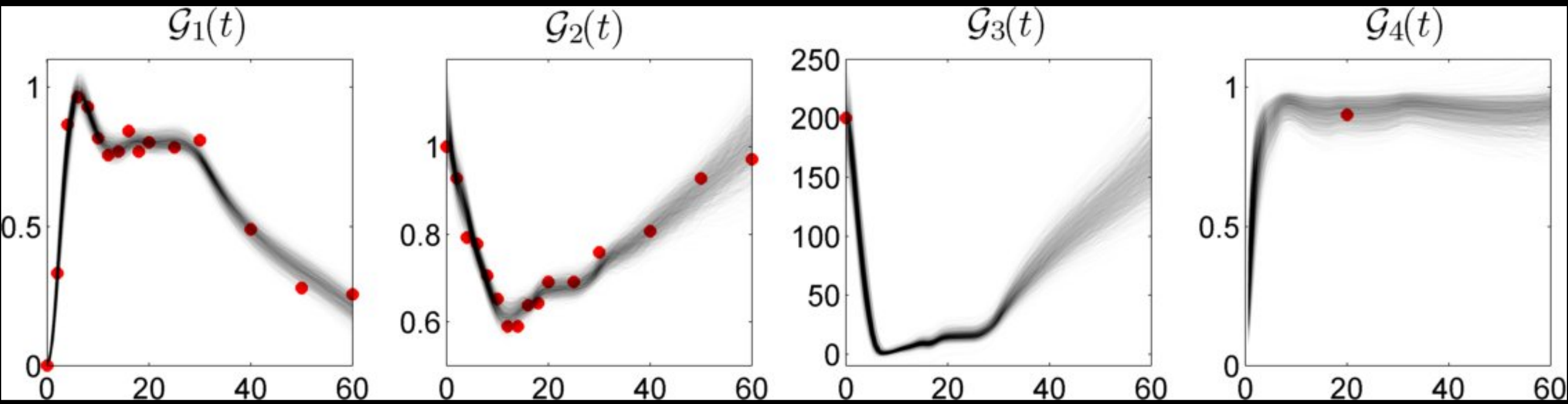
- Single parameter Lorenz estimation problem



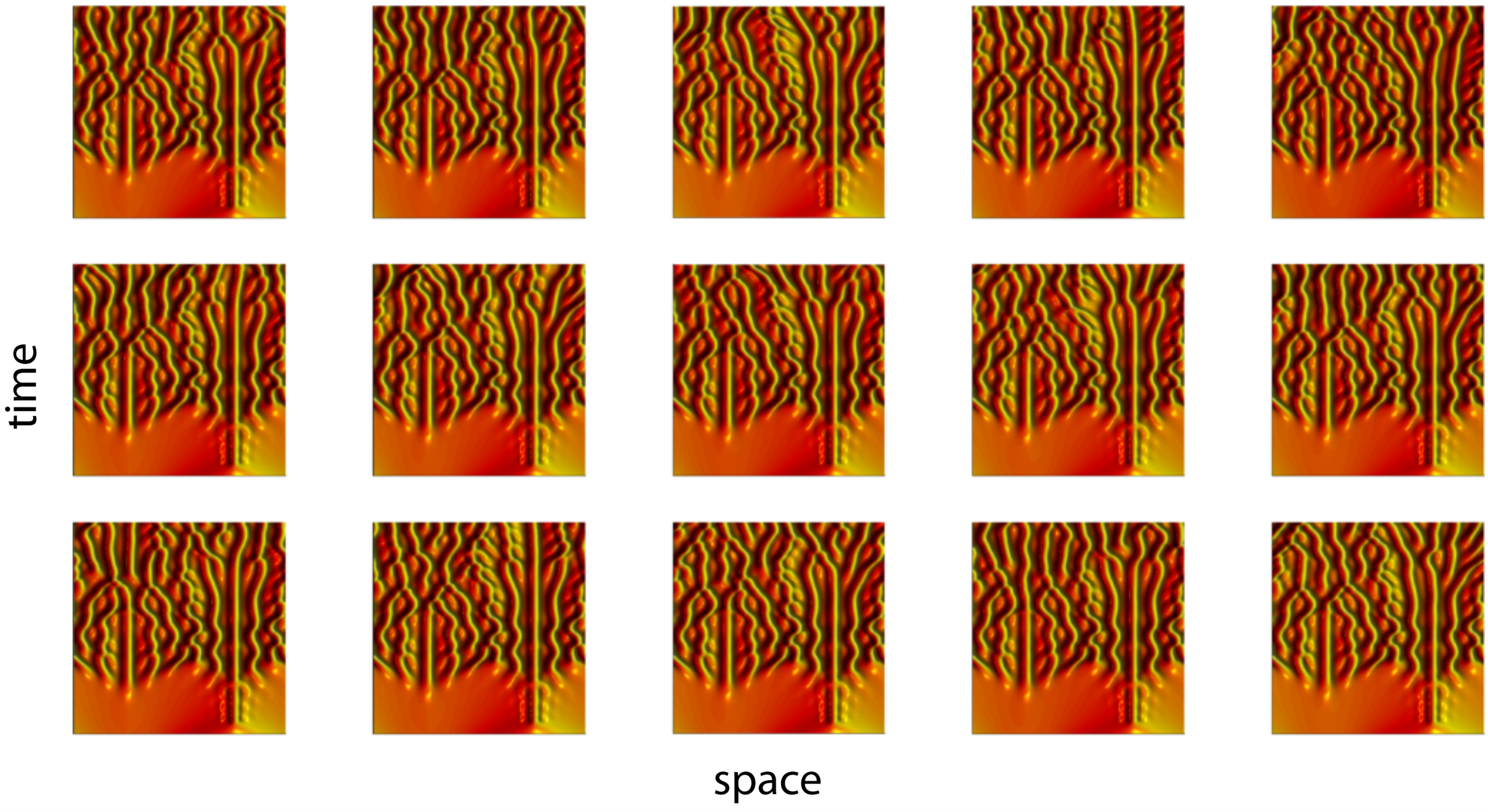


Jak-Stat Solution Posterior

- delay differential equation
- Considering the noise in the data (and its impact on parameter uncertainty) and probabilistically quantifying the uncertainty in the solver

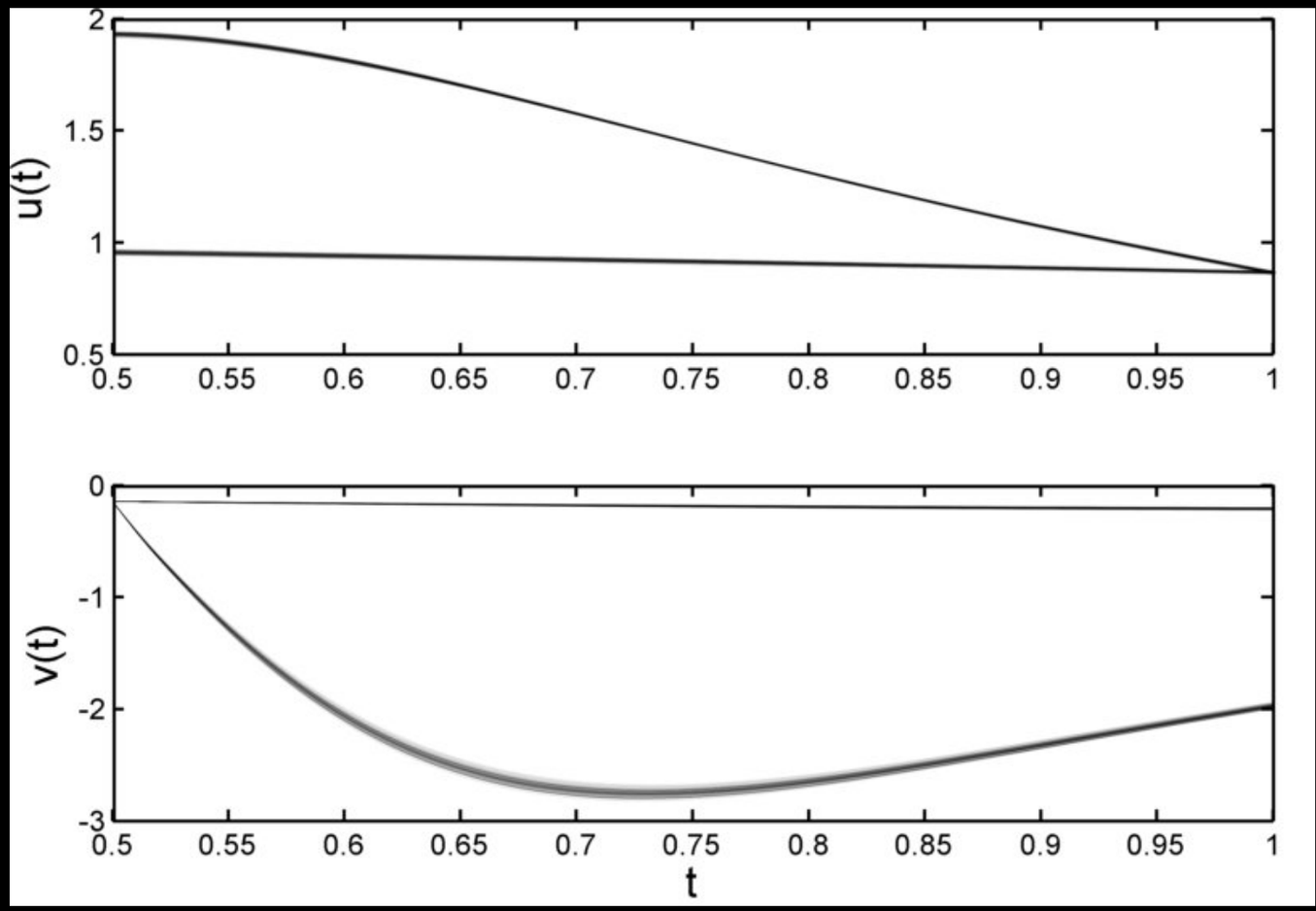


15 PDE solutions



Mixed Boundary Value Problems

Probabilistic methods make ideal solvers for situations subject to multiplicity



Summary

- Accounting for solver error gives a completely probabilistic description of inference.
- Bonus: probabilistic descriptions of a model can smooth out the likelihood for use in parameter estimation!

http://arxiv.org/abs/1306.2365

Next steps

- Implications: Accounting for solution uncertainty could alter the way we look at predictive distributions of long term events like extinctions.
- Functional distributions smooth likelihoods and reduce bias in parameter estimation
- Using a probabilistic solver, what does this mean for bifurcation estimation or stability analysis?
- How does one show uncertainty quantification nicely for a pde?

http://arxiv.org/abs/1306.2365