Probabilistic Numerics for Ordinary Differential Equations

SIAM UQ 2024

Nathanael Bosch

29. February 2024

Eberhard Karls Universität Tübingen

imprs-is

erc some of the presented work is supported by the European Research Council.

Background

- Ordinary differential equations and how to solve them

Background

- Ordinary differential equations and how to solve them

Central statement: **ODE solving is state estimation**

- "ODE filters": **How to solve ODEs with extended Kalman filtering and smoothing**

Background

- Ordinary differential equations and how to solve them

Central statement: **ODE solving is state estimation**

- "ODE filters": **How to solve ODEs with extended Kalman filtering and smoothing**

Fun with ODE filters

- Generalizing ODE filters to other related problems (higher-order ODEs, DAEs, …)
- ODE filters for parameter inference

Background: **Ordinary Differential Equations and how to solve them**

Background: Ordinary Differential Equations and how to solve them

Numerical ODE solvers try to estimate an unknown function by evaluating the vector field

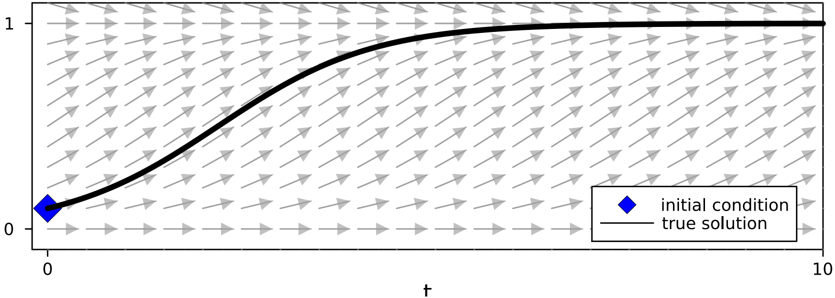
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\dot{y}(t) = f\left(y(t), t\right)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: "Find y".

- **Simple example**: Logistic ODE

$$\dot{y}(t) = y(t)\left(1 - y(t)\right), \qquad t \in [0, 10], \qquad y(0) = 0.1.$$



Background: Ordinary Differential Equations and how to solve them

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Numerical ODE solvers try to estimate an unknown function by evaluating the vector field

$$\dot{y}(t) = f\left(y(t), t\right)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: "Find y".

Numerical ODE solvers:

- Forward Euler:
  $$\hat{y}(t + h) = \hat{y}(t) + hf(\hat{y}(t), t)$$

$$\dot{y}(t) = f\left(y(t), t\right)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: "Find y".

Numerical ODE solvers:

- Forward Euler:
  $$\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t), t)$$
- Backward Euler:
  $$\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t+h), t+h)$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\dot{y}(t) = f\left(y(t), t\right)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: “Find y”.

Numerical ODE solvers:

- Forward Euler:
  $$\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t), t)$$
- Backward Euler:
  $$\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t+h), t+h)$$
- Runge–Kutta:
  $$\hat{y}(t+h) = \hat{y}(t) + h \sum_{i=1}^{s} b_i f(\tilde{y}_i, t + c_i h)$$

Background: Ordinary Differential Equations and how to solve them

Numerical ODE solvers try to estimate an unknown function by evaluating the vector field

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\dot{y}(t) = f\left(y(t), t\right)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: "Find y".

Numerical ODE solvers:

- Forward Euler:
  $$\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t), t)$$
- Backward Euler:
  $$\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t+h), t+h)$$
- Runge–Kutta:
  $$\hat{y}(t+h) = \hat{y}(t) + h\sum_{i=1}^{s} b_i f(\tilde{y}_i, t + c_i h)$$
- Multistep:
  $$\hat{y}(t+h) = \hat{y}(t) + h\sum_{i=0}^{s-1} b_i f(\hat{y}(t-ih), t-ih)$$

Numerical ODE solvers try to estimate an unknown function by evaluating the vector field

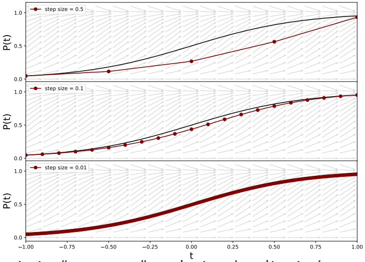
$$\dot{y}(t) = f(y(t), t)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: "Find y".

Numerical ODE solvers:

- Forward Euler:
  $\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t), t)$
- Backward Euler:
  $\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t+h), t+h)$
- Runge–Kutta:
  $\hat{y}(t+h) = \hat{y}(t) + h\sum_{i=1}^{s} b_i f(\tilde{y}_i, t + c_i h)$
- Multistep:
  $\hat{y}(t+h) = \hat{y}(t) + h\sum_{i=0}^{s-1} b_i f(\hat{y}(t - ih), t - ih)$

Forward Euler for different step sizes:



$\Rightarrow$ It is "correct" only in the limit $h \to 0$!

Background: Ordinary Differential Equations and how to solve them

Numerical ODE solvers try to estimate an unknown function by evaluating the vector field

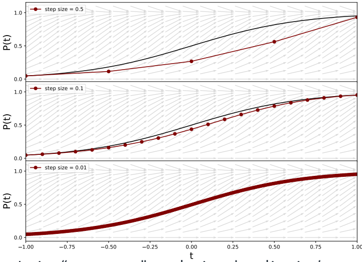
$$\dot{y}(t) = f\left(y(t), t\right)$$

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$, and initial value $y(0) = y_0$. Goal: "Find y".

Numerical ODE solvers:

- Forward Euler:
  $\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t), t)$
- Backward Euler:
  $\hat{y}(t+h) = \hat{y}(t) + hf(\hat{y}(t+h), t+h)$
- Runge–Kutta:
  $\hat{y}(t+h) = \hat{y}(t) + h \sum_{i=1}^{s} b_i f(\tilde{y}_i, t + c_i h)$
- Multistep:
  $\hat{y}(t+h) = \hat{y}(t) + h \sum_{i=0}^{s-1} b_i f(\hat{y}(t - ih), t - ih)$

Forward Euler for different step sizes:



$\Rightarrow$ It is "correct" only in the limit $h \rightarrow 0$!

Numerical ODE solvers **estimate** $y(t)$ *by evaluating f on a discrete set of points.*

Probabilistic numerical ODE solvers

or "How to treat ODEs as a Bayesian state estimation problem"

Probabilistic Numerical ODE Solvers

How to treat ODEs as the state estimation problem that they really are

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f\left(y(t_n), t_n\right)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f(y(t_n), t_n)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

- **Prior:** $y(t) \sim \mathcal{GP}$

Probabilistic Numerical ODE Solvers

How to treat ODEs as the state estimation problem that they really are

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f\left(y(t_n), t_n\right)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

- **Prior:** $y(t) \sim \mathcal{GP}$ a Gauss–Markov process

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f\left(y(t_n), t_n\right)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

- **Prior:** $y(t) \sim \mathcal{GP}$ a Gauss–Markov process with state-space representation $x(t)$:

$$x(0) \sim \mathcal{N}(\mu_0^-, \Sigma_0^-),$$
$$\mathrm{d}x(t) = Fx(t)\mathrm{d}t + \sigma\Gamma\mathrm{d}w(t),$$
$$y^{(m)}(t) = E_m x(t), \quad m = 1, \dots, \nu.$$

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f(y(t_n), t_n)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

- **Prior:** $y(t) \sim \mathcal{GP}$ a Gauss–Markov process with state-space representation $x(t)$:

$$\begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ \mathrm{d}x(t) &= Fx(t)\mathrm{d}t + \sigma\Gamma\mathrm{d}w(t), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned} \quad \Rightarrow \quad \begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ x(t_{i+1}) \mid x(t_i) &\sim \mathcal{N}\left(A(\Delta_i)x(t), \sigma^2 Q(\Delta_i)\right), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned}$$

where $\Delta_i := t_{i+1} - t_i$, and (A, Q) can be computed from (F, Γ).

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f\left(y(t_n), t_n\right)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^N$.

- **Prior:** $y(t) \sim \mathcal{GP}$ a Gauss–Markov process with state-space representation $x(t)$:

$$\begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ \mathrm{d}x(t) &= Fx(t)\mathrm{d}t + \sigma\Gamma\mathrm{d}w(t), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned} \quad \Rightarrow \quad \begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ x(t_{i+1}) \mid x(t_i) &\sim \mathcal{N}\left(A(\Delta_i)x(t), \sigma^2 Q(\Delta_i)\right), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned}$$

  where $\Delta_i := t_{i+1} - t_i$, and (A, Q) can be computed from (F, Γ).
- **Likelihood:** (aka “observation model” or “information operator”)

$$z_0 = E_0 x(0) - y_0 = 0, \qquad \& \qquad z(t_n) = E_1 x(t_n) - f(E_0 x(t_n), t_n) = 0.$$

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f(y(t_n), t_n)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

- **Prior:** $y(t) \sim \mathcal{GP}$ a Gauss–Markov process with state-space representation $x(t)$:

$$\begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ \mathrm{d}x(t) &= Fx(t)\mathrm{d}t + \sigma\Gamma \mathrm{d}w(t), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned} \quad \Rightarrow \quad \begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ x(t_{i+1}) \mid x(t_i) &\sim \mathcal{N}\left(A(\Delta_i)x(t), \sigma^2 Q(\Delta_i)\right), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned}$$

  where $\Delta_i := t_{i+1} - t_i$, and (A, Q) can be computed from (F, Γ).
- **Likelihood:** (aka “observation model” or “information operator”)

$$z_0 = E_0 x(0) - y_0 = 0, \qquad \& \qquad z(t_n) = E_1 x(t_n) - f(E_0 x(t_n), t_n) = 0.$$

- **Inference:**

$$p\left(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) = f\left(y(t_n), t_n\right)\}_{n=1}^{N}\right)$$

with vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, initial value y_0, and time discretization $\{t_n\}_{n=1}^{N}$.

- **Prior:** $y(t) \sim \mathcal{GP}$ a Gauss–Markov process with state-space representation $x(t)$:

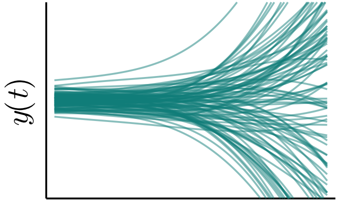
$$\begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ \mathrm{d}x(t) &= Fx(t)\mathrm{d}t + \sigma\Gamma\mathrm{d}w(t), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned} \quad \Rightarrow \quad \begin{aligned} x(0) &\sim \mathcal{N}(\mu_0^-, \Sigma_0^-), \\ x(t_{i+1}) \mid x(t_i) &\sim \mathcal{N}\left(A(\Delta_i)x(t), \sigma^2 Q(\Delta_i)\right), \\ y^{(m)}(t) &= E_m x(t), \quad m = 1, \dots, \nu. \end{aligned}$$

  where $\Delta_i := t_{i+1} - t_i$, and (A, Q) can be computed from (F, Γ).
- **Likelihood:** (aka “observation model” or “information operator”)

$$z_0 = E_0 x(0) - y_0 = 0, \qquad \& \qquad z(t_n) = E_1 x(t_n) - f(E_0 x(t_n), t_n) = 0.$$

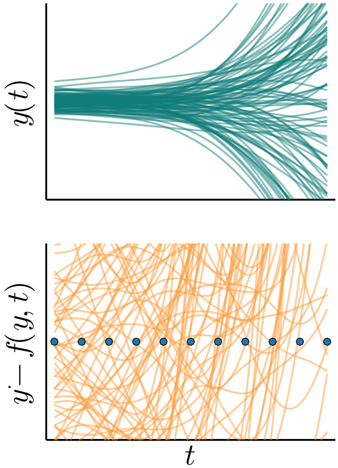
- **Inference:** Extended Kalman filter/smoother (or other Bayesian filtering and smoothing methods).

Prior

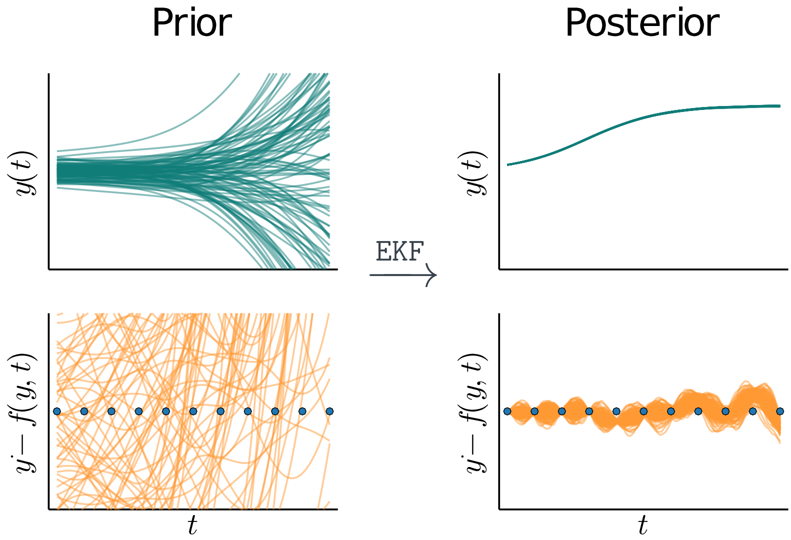


Probabilistic Numerical ODE Solvers in pictures

Prior

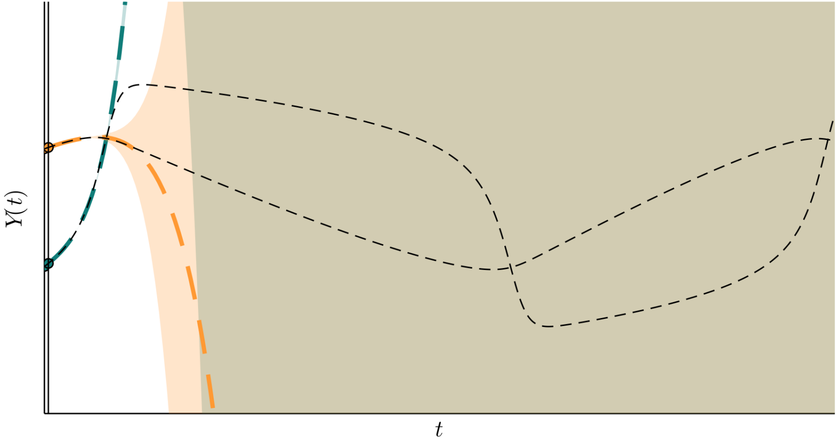


Probabilistic Numerical ODE Solvers in pictures



Probabilistic Numerical ODE Solvers in action

Fixed steps — the vanilla way as introduced so far



Probabilistic Numerical ODE Solvers in code

We can solve ODEs with basically just an extended Kalman filter

Algorithm The extended Kalman ODE filter

procedure EXTENDED KALMAN ODE FILTER$((\mu_0^-, \Sigma_0^-), (A, Q), (f, x_0), \{t_i\}_{i=1}^N)$
    $\mu_0, \Sigma_0 \leftarrow \text{KF\_UPDATE}(\mu_0^-, \Sigma_0^-, E_0, 0_{d \times d}, x_0)$ // Initial update to fit the initial value
    for $k \in \{1, \dots, N\}$ **do**
        $h_k \leftarrow t_k - t_{k-1}$ // Step size
        $\mu_k^-, \Sigma_k^- \leftarrow \text{KF\_PREDICT}(\mu_{k-1}, \Sigma_{k-1}, A(h_k), Q(h_k))$ // Kalman filter prediction
        $m_k(X) := E_1 X - f(E_0 X, t_k)$ // Define the non-linear observation model
        $\mu_k, \Sigma_k \leftarrow \text{EKF\_UPDATE}(\mu_k^-, \Sigma_k^-, m_k, 0_{d \times d}, \mathbf{0}_d)$ // Extended Kalman filter update
    end for
    return $(\mu_k, \Sigma_k)_{k=1}^N$
end procedure

EXTENDED KALMAN ODE SMOOTHER: Just run a RTS smoother after the filter!

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]
  - Step-size adaptation and calibration: [Bosch et al., 2021]

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]
  - Step-size adaptation and calibration: [Bosch et al., 2021]
  - Parallel-in-time formulation [Bosch et al., 2023a]

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]
  - Step-size adaptation and calibration: [Bosch et al., 2021]
  - Parallel-in-time formulation [Bosch et al., 2023a]
- More related differential equation problems:
  - Higher-order ODEs, DAEs, Hamiltonian systems [Bosch et al., 2022]
  - Boundary value problems (BVPs) [Krämer and Hennig, 2021]
  - Partial differential equations (PDEs) via method of lines [Krämer et al., 2022]

The state of filtering-based probabilistic numerical ODE solvers

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]
  - Step-size adaptation and calibration: [Bosch et al., 2021]
  - Parallel-in-time formulation [Bosch et al., 2023a]
- More related differential equation problems:
  - Higher-order ODEs, DAEs, Hamiltonian systems [Bosch et al., 2022]
  - Boundary value problems (BVPs) [Krämer and Hennig, 2021]
  - Partial differential equations (PDEs) via method of lines [Krämer et al., 2022]
- Inverse problems
  - Parameter inference in ODEs with ODE filters [Tronarp et al., 2022]
  - Efficient latent force inference [Schmidt et al., 2021]

The state of filtering-based probabilistic numerical ODE solvers

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]
  - Step-size adaptation and calibration: [Bosch et al., 2021]
  - Parallel-in-time formulation [Bosch et al., 2023a]
- More related differential equation problems:
  - Higher-order ODEs, DAEs, Hamiltonian systems [Bosch et al., 2022]
  - Boundary value problems (BVPs) [Krämer and Hennig, 2021]
  - Partial differential equations (PDEs) via method of lines [Krämer et al., 2022]
- Inverse problems
  - Parameter inference in ODEs with ODE filters [Tronarp et al., 2022]
  - Efficient latent force inference [Schmidt et al., 2021]

Probabilistic Numerics: Computation as Machine Learning

Philipp Hennig, Michael A. Osborne, Hans P. Kersting, 2022

The state of filtering-based probabilistic numerical ODE solvers

- Properties and features:
  - Polynomial convergence rates [Kersting et al., 2020, Tronarp et al., 2021]
  - A-stability [Tronarp et al., 2019]
  - L-stable probabilistic exponential integrators [Bosch et al., 2023b]
  - Complexity: $\mathcal{O}(d^3)$ for the A-stable semi-implicit method, $\mathcal{O}(d)$ for an explicit method with coarser covariances [Krämer et al., 2022]
  - Step-size adaptation and calibration: [Bosch et al., 2021]
  - Parallel-in-time formulation [Bosch et al., 2023a]
- More related differential equation problems:
  - Higher-order ODEs, DAEs, Hamiltonian systems [Bosch et al., 2022]
  - Boundary value problems (BVPs) [Krämer and Hennig, 2021]
  - Partial differential equations (PDEs) via method of lines [Krämer et al., 2022]
- Inverse problems
  - Parameter inference in ODEs with ODE filters [Tronarp et al., 2022]
  - Efficient latent force inference [Schmidt et al., 2021]

Probabilistic Numerics: Computation as Machine Learning

Philipp Hennig, Michael A. Osborne, Hans P. Kersting, 2022

Flexible Information Operators

or: *"How to solve other problems than ODEs with essentially the same algorithm as before"*

Flexible Information Operators

or: *"How to solve other problems than ODEs with essentially the same algorithm as before"*
(it's all just likelihood models)

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with first-order ODE

$$\dot{y}(t) = f(y(t), t), \quad y(0) = y_0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_1 x(t_i) - f(E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with second-order ODE

$$\ddot{y}(t) = f(\dot{y}(t), y(t), t), \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_1 x(t_i) - f(E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

Extending ODE filters to other related differential equation problems

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

ODE filters can solve much more than the ODEs that we saw so far!

[Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with second-order ODE

$$\ddot{y}(t) = f(\dot{y}(t), y(t), t), \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_2 x(t_i) - f(E_1 x(t_i), E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$
Initial derivative likelihood:	$z_1^{\text{init}} \mid x(0) \sim \delta\left(z_1^{\text{init}}; E_1 x(0)\right),$	$z_1^{\text{init}} \triangleq \dot{y}_0$

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with first-order ODE and conserved quantities

$$\dot{y}(t) = f(y(t), t), \quad y(0) = y_0. \quad g(y(t), \dot{y}(t)) = 0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_1 x(t_i) - f(E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

Numerical problems setting: Initial value problem with first-order ODE and conserved quantities

$$\dot{y}(t) = f(y(t), t), \quad y(0) = y_0. \quad g(y(t), \dot{y}(t)) = 0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_1 x(t_i) - f(E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Conservation law likelihood:	$z_i^c(t_i) \mid z(t_i) \sim \delta\left(z_i^c(t_i); g(E_0 x(t), E_1 x(t))\right),$	$z_i^c \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with second-order ODE and conserved quantities

$$\ddot{y}(t) = f(\dot{y}(t), y(t), t), \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0. \quad g(y(t), \dot{y}(t)) = 0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_2 x(t_i) - f(E_1 x(t_i), E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Conservation law likelihood:	$z_i^c(t_i) \mid z(t_i) \sim \delta\left(z_i^c(t_i); g(E_0 x(t), E_1 x(t))\right),$	$z_i^c \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$
Initial derivative likelihood:		

Extending ODE filters to other related differential equation problems

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

ODE filters can solve much more than the ODEs that we saw so far!

[Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with second-order ODE and conserved quantities

$\ddot{y}(t)$ … $= 0.$

This leads to the **pro**

Initial distr

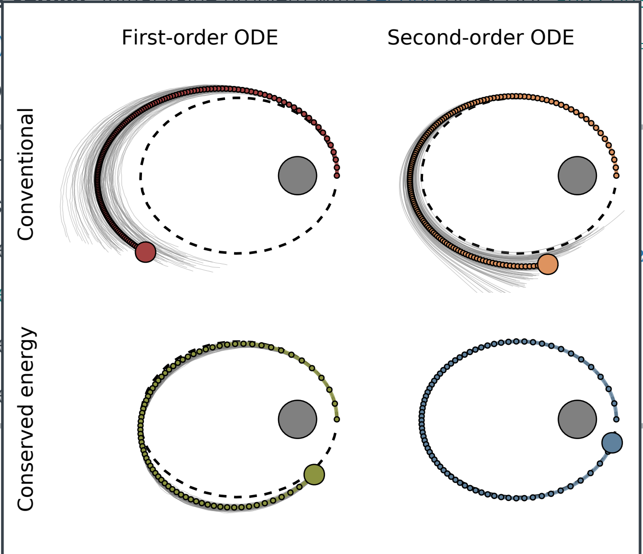
Prior / dynamics

ODE lik … $x(t_i) - f(E_1 x(t_i), E_0 x(t_i), t_i)$

Conservation law lik

Initial value lik

Initial derivative lik



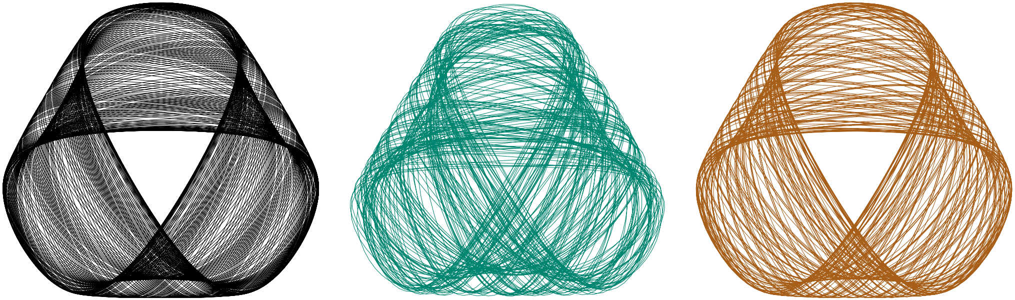
Extending ODE filters to other related differential equation problems

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with second-order ODE and conserved quantities

$$\ddot{y}(t) = f(\dot{y}(t), y(t), t), \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0. \quad g(y(t), \dot{y}(t)) = 0.$$



$(t_i), t_i)$

Numerical problems setting: Initial value problem with *differential-algebraic equation* (DAE)

$$0 = F\left(\dot{y}(t), y(t), t\right), \quad y(0) = y_0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_1 x(t_i) - f(E_0 x(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Initial value problem with *differential-algebraic equation* (DAE)

$$0 = F\left(\dot{y}(t), y(t), t\right), \quad y(0) = y_0.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
DAE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); F\left(E_1 x(t_i), E_0 x(t_i), t_i\right)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Boundary value problem (BVP) with first-order ODE

$$\dot{y}(t) = f(y(t), t), \quad Ly(0) = y_0, \quad Ry(T) = y_T.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); \right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

Numerical problems setting: Boundary value problem (BVP) with first-order ODE

$$\dot{y}(t) = f(y(t), t), \quad Ly(0) = y_0, \quad Ry(T) = y_T.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); \right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; LE_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$
Boundary value likelihood:	$z_1^{\text{R}} \mid x(T) \sim \delta\left(z_1^{\text{R}}; RE_0 x(T)\right),$	$z_1^{\text{init}} \triangleq y_T$

Extending ODE filters to other related differential equation problems

ODE filters can solve much more than the ODEs that we saw so far! [Bosch et al., 2022, Krämer and Hennig, 2021]

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Numerical problems setting: Boundary value problem (BVP) with first-order ODE

$$\dot{y}(t) = f(y(t), t), \quad Ly(0) = y_0, \quad Ry(T) = y_T.$$

This leads to the **probabilistic state estimation problem:**

Initial distribution:	$x(0) \sim \mathcal{N}\left(x(0); \mu_0^-, \Sigma_0^-\right)$	
Prior / dynamics model:	$x(t+h) \mid x(t) \sim \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right)$	
ODE likelihood:	$z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); \right),$	$z_i \triangleq 0$
Initial value likelihood:	$z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; LE_0 x(0)\right),$	$z^{\text{init}} \triangleq y_0$
Boundary value likelihood:	$z_1^{\text{R}} \mid x(T) \sim \delta\left(z_1^{\text{R}}; RE_0 x(T)\right),$	$z_1^{\text{init}} \triangleq y_T$

The measurement model provides a very flexible way to easily encode desired properties.
But it's all just Bayesian state estimation!

Probabilistic Numerics for ODE Parameter Inference

Using the ODE solution as a “physics-enhanced” prior for regression

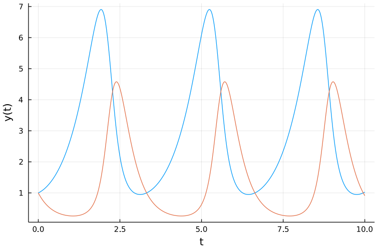
“Forward” and “Inverse” Problems

Going from formula to plot, or from plot to formula

Forward Problem

$$\dot{y}_\theta = f_\theta(y_\theta, t) \qquad y_\theta(t_0) = y_0(\theta).$$

solve $\Longrightarrow$



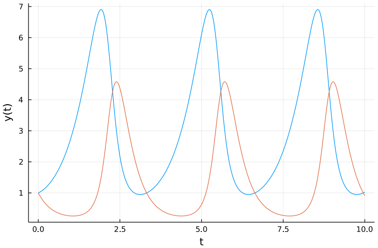
“Forward” and “Inverse” Problems

Going from formula to plot, or from plot to formula

Forward Problem

$$\dot{y}_\theta = f_\theta(y_\theta, t) \qquad y_\theta(t_0) = y_0(\theta).$$

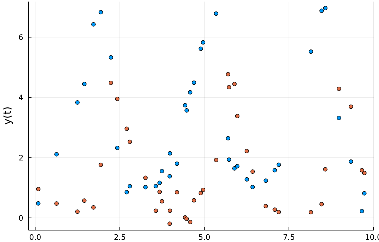
solve $\Longrightarrow$



Inverse Problem

$$\hat{\theta} = \arg\max_\theta p\left(\mathcal{D} \mid \theta\right)$$

find $\Longleftarrow$



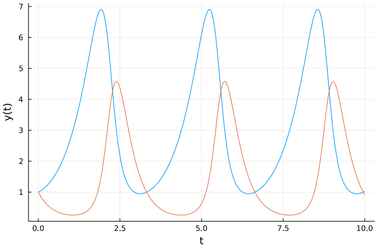
“Forward” and “Inverse” Problems

Going from formula to plot, or from plot to formula

Forward Problem

$$\dot{y}_\theta = f_\theta(y_\theta, t) \qquad y_\theta(t_0) = y_0(\theta).$$

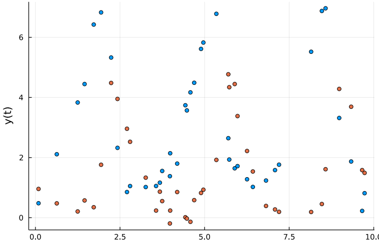
solve $\Longrightarrow$



Inverse Problem

$$\hat{\theta} = \arg\max_\theta p\left(\mathcal{D} \mid \theta\right)$$

find $\Longleftarrow$



Problem: The *marginal likelihood* $p(\mathcal{D} \mid \theta) = \prod \mathcal{N}\left(u(t); y_\theta(t), R_\theta\right)$ is intractable. (because the true ODE solution is intractable!)

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$
2. **Gradient Matching**
   - (i) Fit a curve $\hat{y}(t)$ *to the data* $\{u(t_i)\}$
   - (ii) Estimate θ by minimizing $\dot{\hat{y}}(t) - f_\theta(\hat{y}(t))$

   Exists in both classic (splines) or probabilistic versions (GPs)

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$
2. **Gradient Matching**
   - (i) Fit a curve $\hat{y}(t)$ *to the data* $\{u(t_i)\}$
   - (ii) Estimate θ by minimizing $\dot{\hat{y}}(t) - f_\theta(\hat{y}(t))$

   Exists in both classic (splines) or probabilistic versions (GPs)
3. **Probabilistic Numerical Integration**

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$
2. **Gradient Matching**
   - (i) Fit a curve $\hat{y}(t)$ *to the data* $\{u(t_i)\}$
   - (ii) Estimate θ by minimizing $\dot{\hat{y}}(t) - f_\theta(\hat{y}(t))$

   Exists in both classic (splines) or probabilistic versions (GPs)
3. **Probabilistic Numerical Integration**
   - $$\widehat{\mathcal{M}}_{PN}(\theta, \kappa) = \int \underbrace{\prod_n \mathcal{N}\left(u(t_n); y(t_n), R_\theta\right)}_{\text{Likelihood}} \cdot \underbrace{\gamma_{PN}\left(y(t_{1:N}) \mid \theta, \kappa\right)}_{\text{PN ODE Solution}} \mathrm{d}y(t_{1:N}) \tag{1}$$

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$
2. **Gradient Matching**
   - (i) Fit a curve $\hat{y}(t)$ *to the data* $\{u(t_i)\}$
   - (ii) Estimate θ by minimizing $\dot{\hat{y}}(t) - f_\theta(\hat{y}(t))$

   Exists in both classic (splines) or probabilistic versions (GPs)
3. **Probabilistic Numerical Integration**
   - $$\widehat{\mathcal{M}}_{PN}(\theta, \kappa) = \int \underbrace{\prod_n \mathcal{N}\left(u(t_n); y(t_n), R_\theta\right)}_{\text{Likelihood}} \cdot \underbrace{\gamma_{PN}\left(y(t_{1:N}) \mid \theta, \kappa\right)}_{\text{PN ODE Solution}} \mathrm{d}y(t_{1:N}) \tag{1}$$
   - (i) *Probabilistically* solve IVP to compute $\gamma_{PN}(y(t) \mid \theta, \kappa)$

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$
2. **Gradient Matching**
   - (i) Fit a curve $\hat{y}(t)$ *to the data* $\{u(t_i)\}$
   - (ii) Estimate θ by minimizing $\dot{\hat{y}}(t) - f_\theta(\hat{y}(t))$

   Exists in both classic (splines) or probabilistic versions (GPs)
3. **Probabilistic Numerical Integration**
   - $$\widehat{\mathcal{M}}_{PN}(\theta, \kappa) = \int \underbrace{\prod_n \mathcal{N}\left(u(t_n); y(t_n), R_\theta\right)}_{\text{Likelihood}} \cdot \underbrace{\gamma_{PN}\left(y(t_{1:N}) \mid \theta, \kappa\right)}_{\text{PN ODE Solution}} \mathrm{d}y(t_{1:N}) \tag{1}$$
   - (i) *Probabilistically* solve IVP to compute $\gamma_{\text{PN}}(y(t) \mid \theta, \kappa)$
   - (ii) Perform Kalman filtering on the data, with γ_{PN} as a "physics-enhanced" **prior**

Context: Between classic integration and gradient matching

We're doing both: Integrating first, then GP regression

1. **Classical Numerical Integration**
   - (i) Solve the IVP to compute $\hat{y}_\theta(t)$
   - (ii) Approximate the marginal likelihood as $\widehat{\mathcal{M}}(\theta) = \prod_n \mathcal{N}(u(t_n); \hat{y}_\theta(t_n), R_\theta)$
   - (iii) Optimize to get $\hat{\theta} = \arg\max \widehat{\mathcal{M}}(\theta)$
2. **Gradient Matching**
   - (i) Fit a curve $\hat{y}(t)$ *to the data* $\{u(t_i)\}$
   - (ii) Estimate θ by minimizing $\dot{\hat{y}}(t) - f_\theta(\hat{y}(t))$

   Exists in both classic (splines) or probabilistic versions (GPs)
3. **Probabilistic Numerical Integration**
   - $$\widehat{\mathcal{M}}_{PN}(\theta, \kappa) = \int \underbrace{\prod_n \mathcal{N}\left(u(t_n); y(t_n), R_\theta\right)}_{\text{Likelihood}} \cdot \underbrace{\gamma_{PN}\left(y(t_{1:N}) \mid \theta, \kappa\right)}_{\text{PN ODE Solution}} \mathrm{d}y(t_{1:N}) \quad (1)$$
   - (i) *Probabilistically* solve IVP to compute $\gamma_{\text{PN}}(y(t) \mid \theta, \kappa)$
   - (ii) Perform Kalman filtering on the data, with γ_{PN} as a "physics-enhanced" **prior**
   - (iii) Optimize the approximate marginal likelihood

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

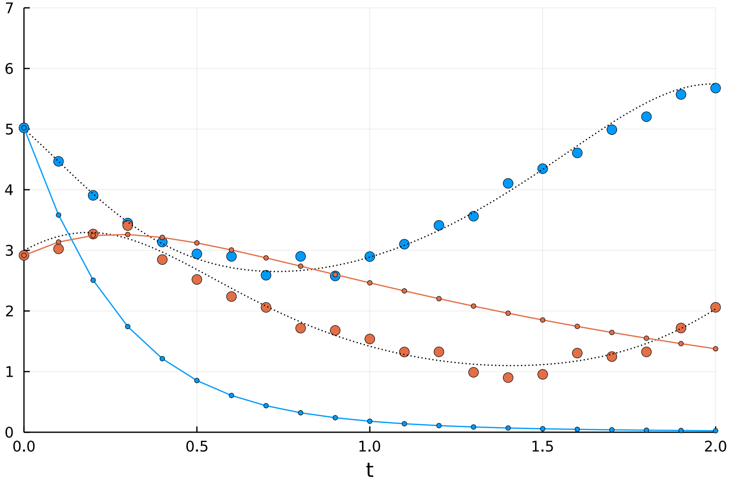


Figure: i=1

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

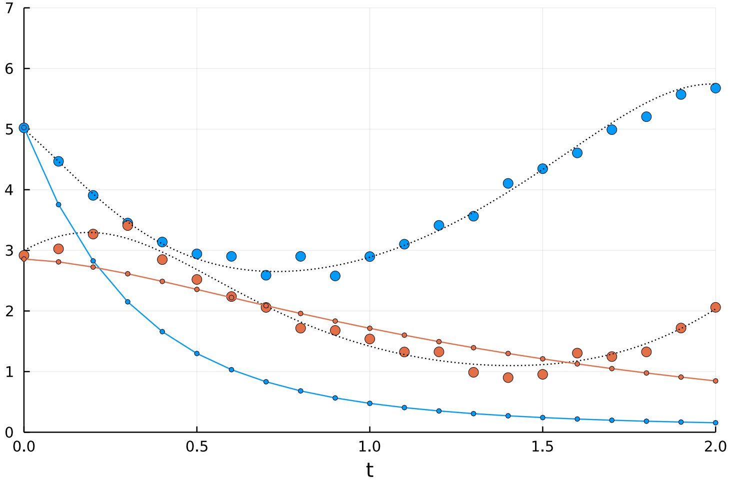


Figure: i=2

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

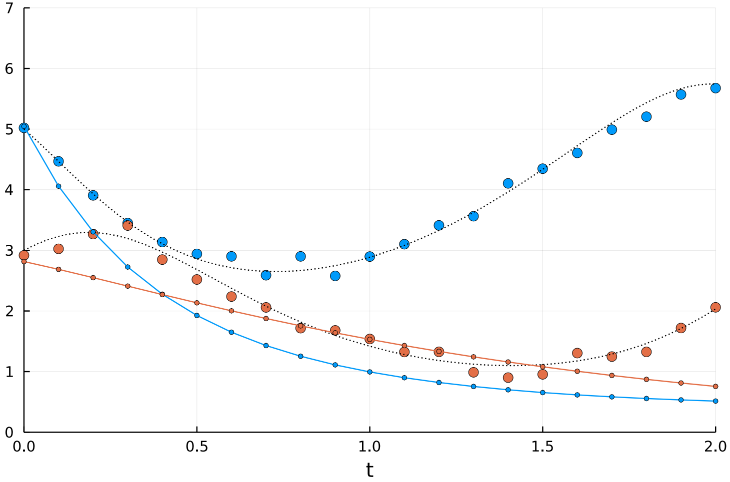


Figure: i=3

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

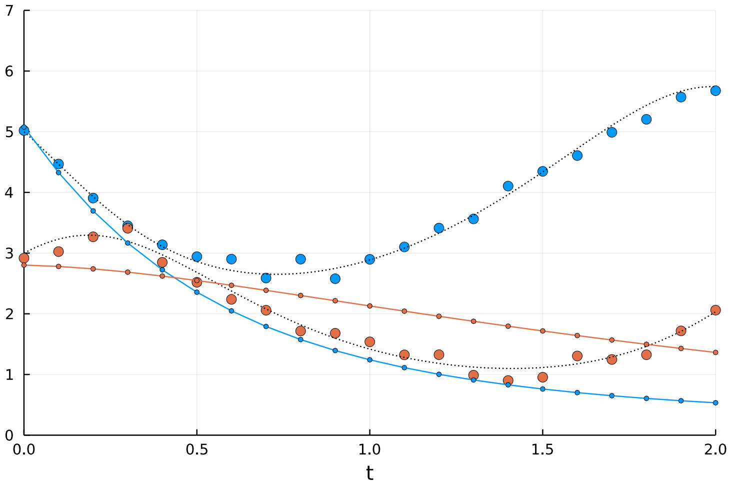


Figure: i=4

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

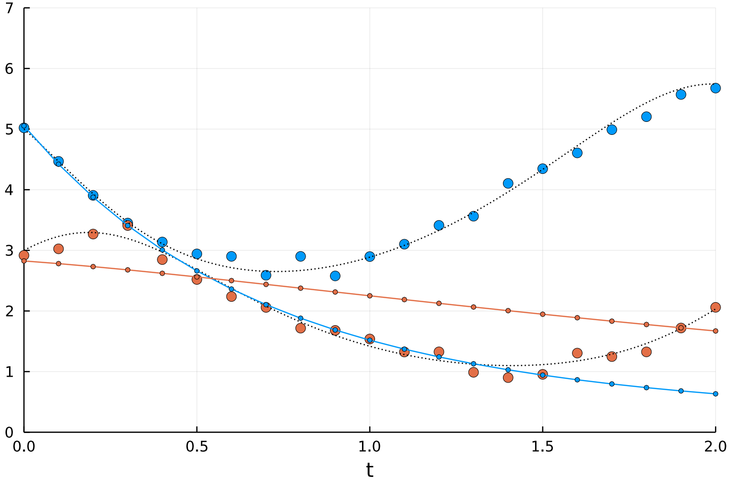


Figure: i=5

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

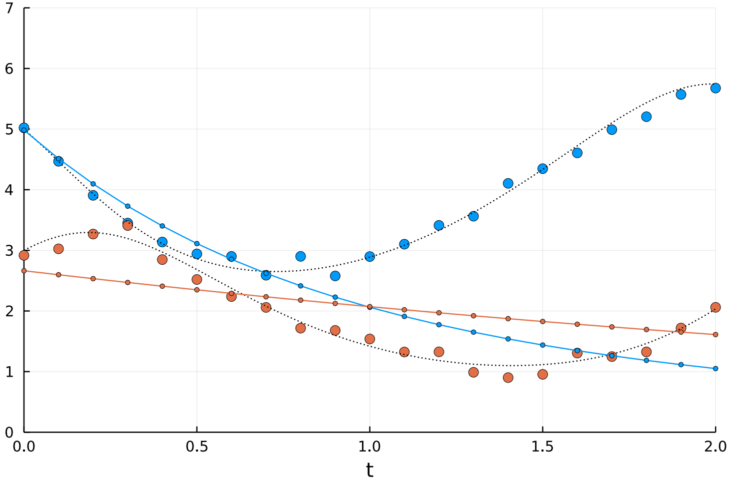


Figure: i=10

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

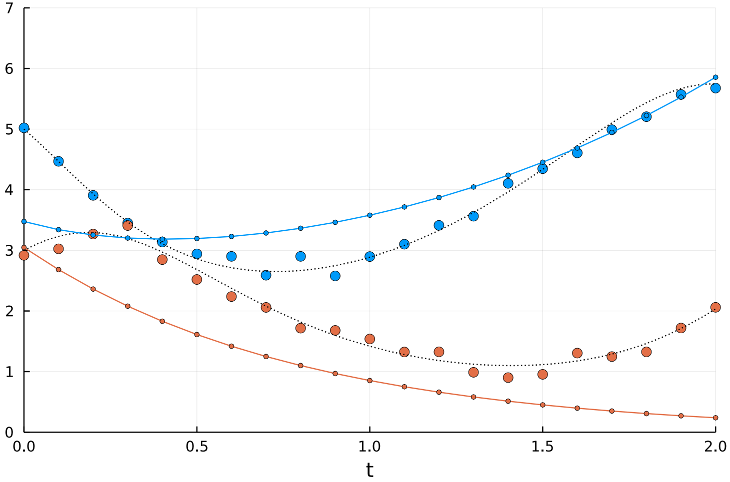


Figure: i=15

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

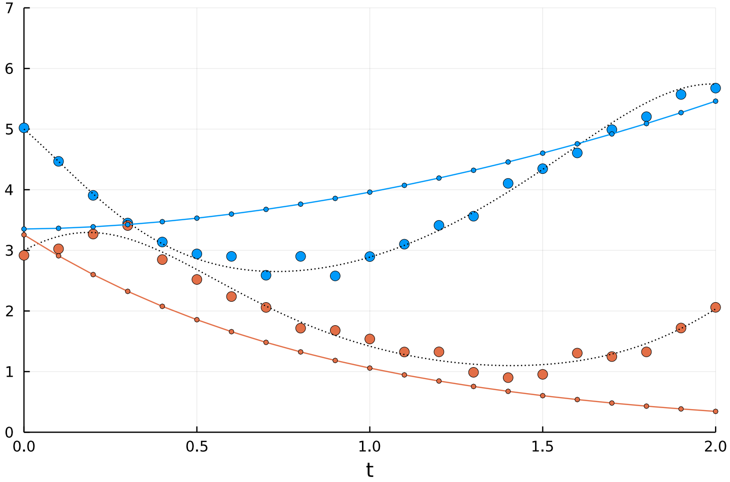


Figure: i=20

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

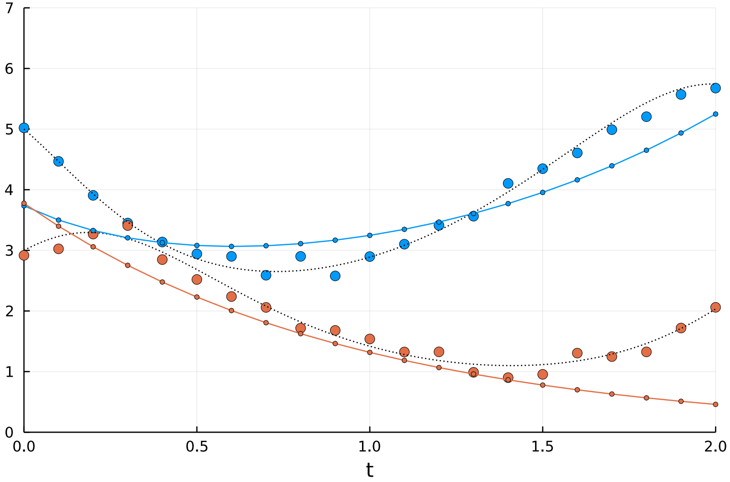


Figure: i=25

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

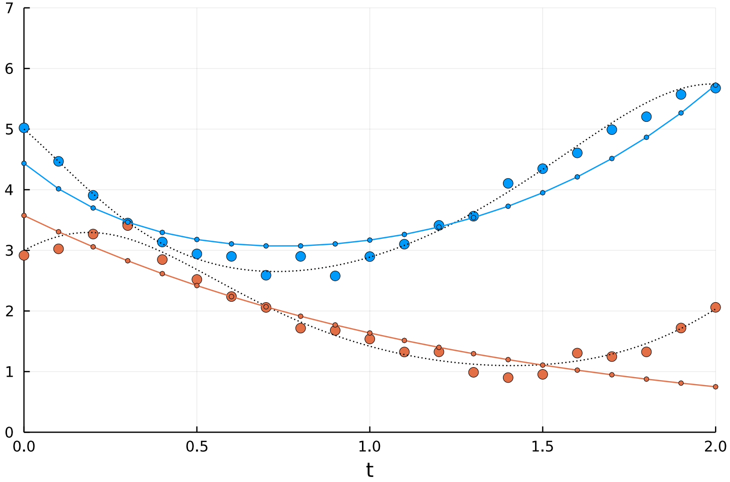


Figure: i=30

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

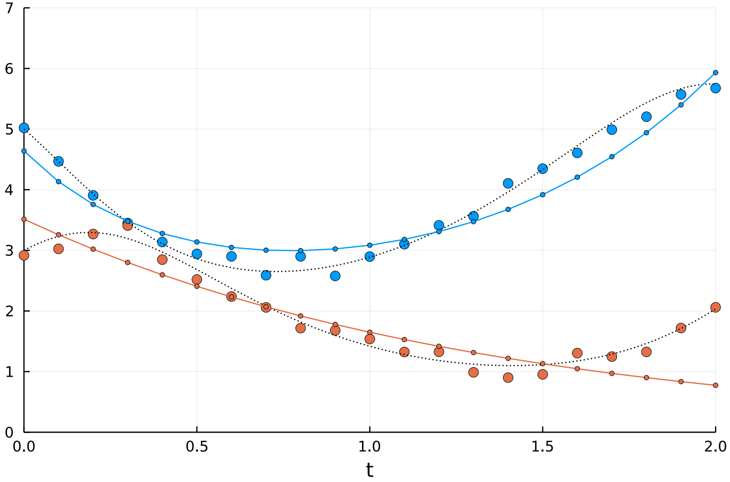


Figure: i=35

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

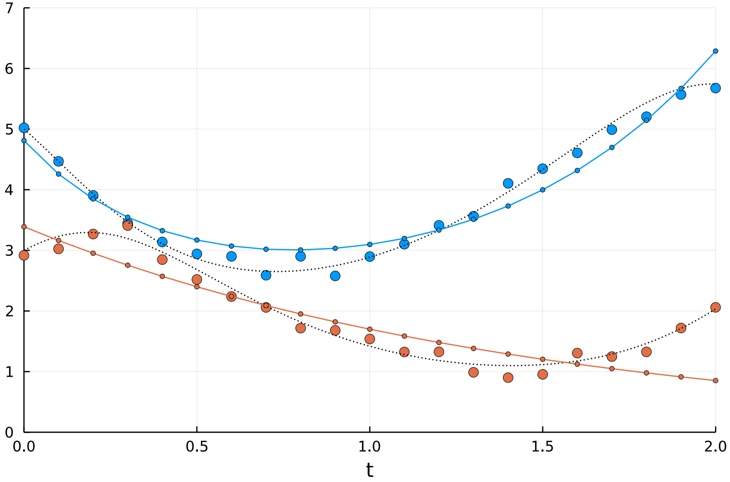


Figure: i=40

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

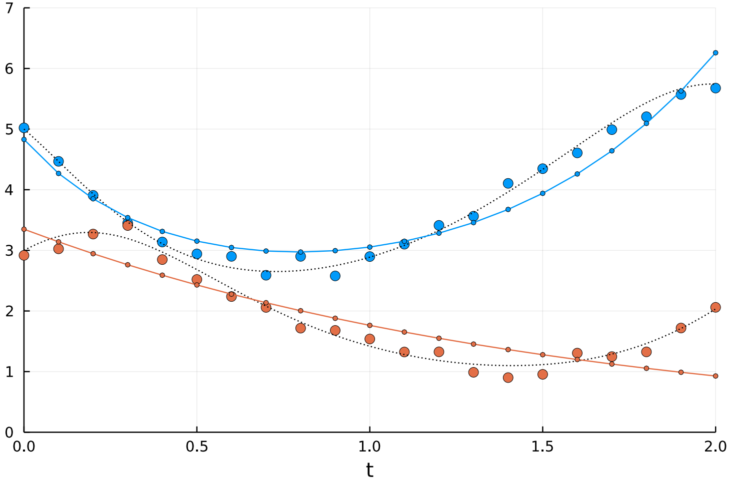


Figure: i=45

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

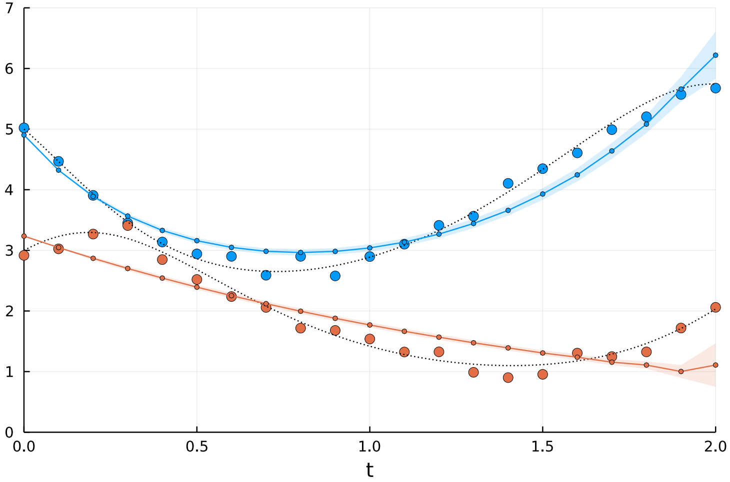


Figure: i=50

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

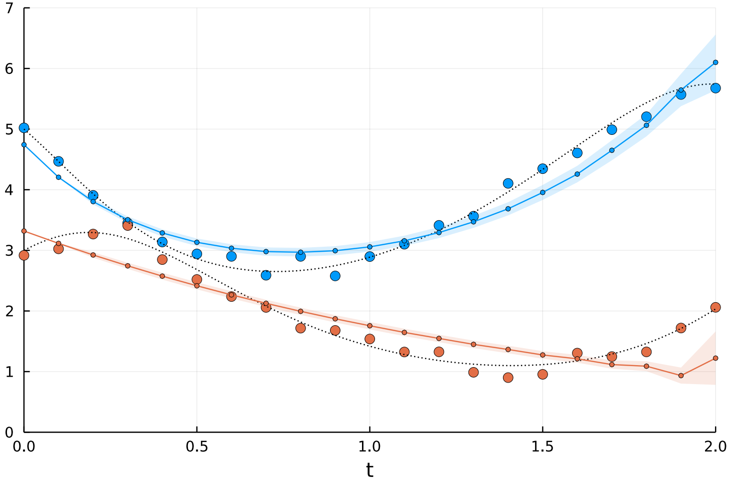


Figure: i=55

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

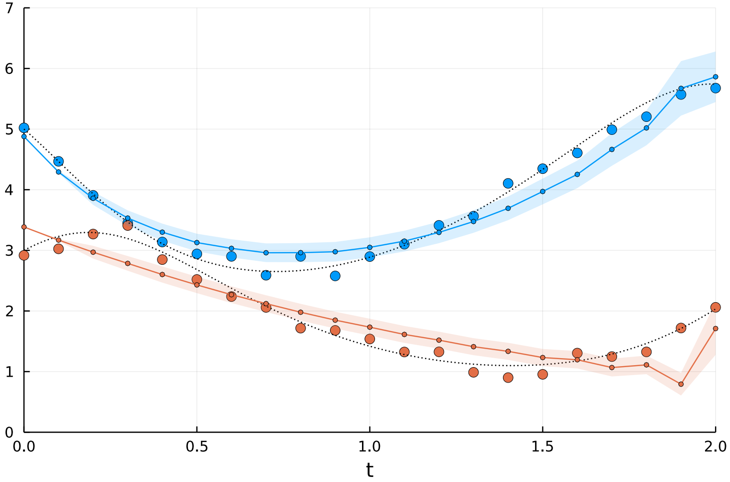


Figure: i=60

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

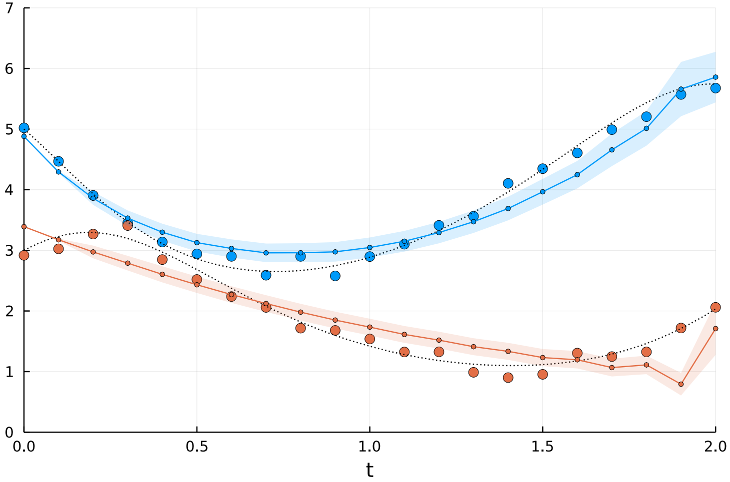


Figure: i=61

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

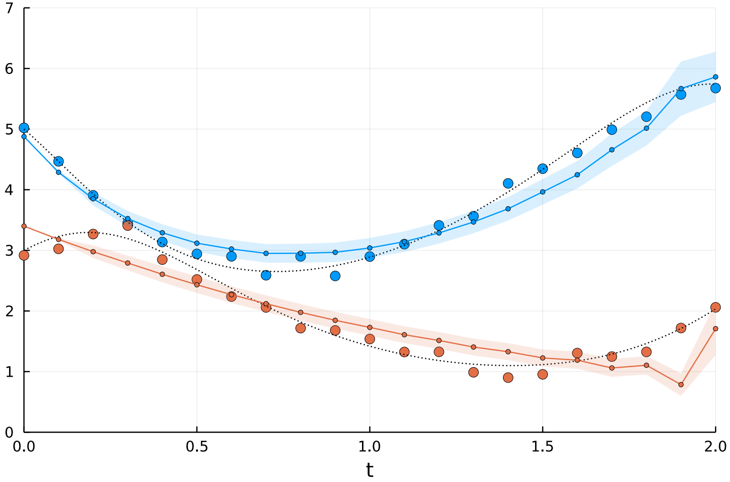


Figure: i=62

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

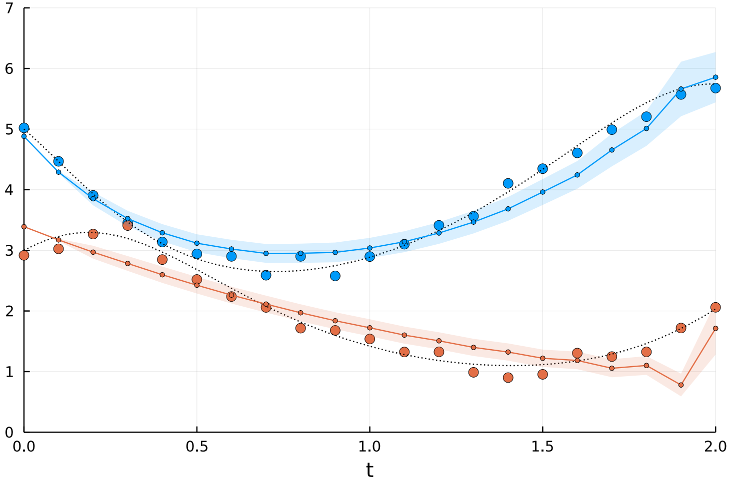


Figure: i=63

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

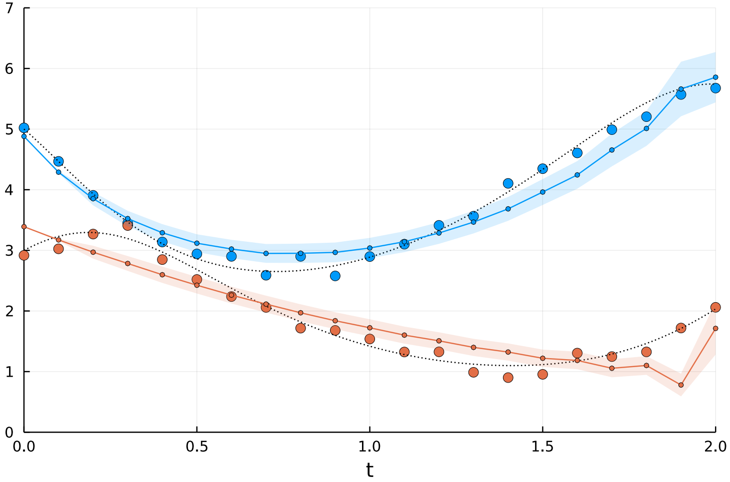


Figure: i=63

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

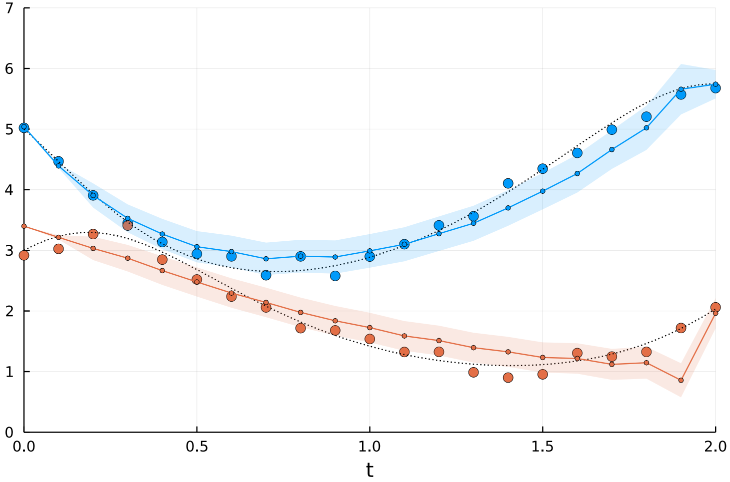


Figure: i=64

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

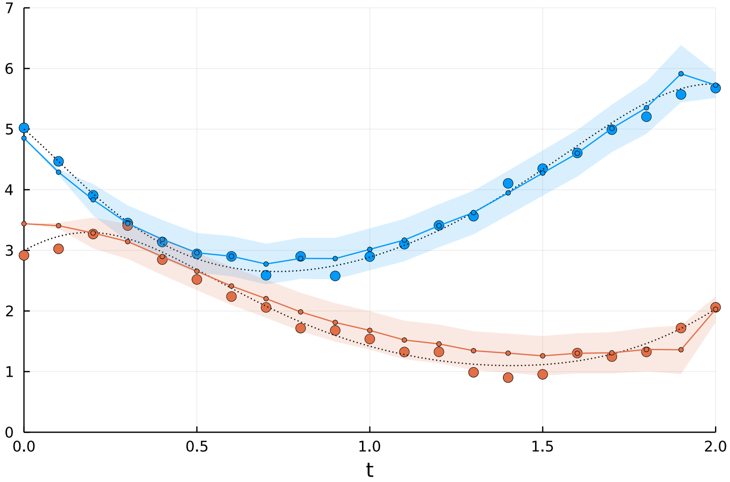


Figure: i=65

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

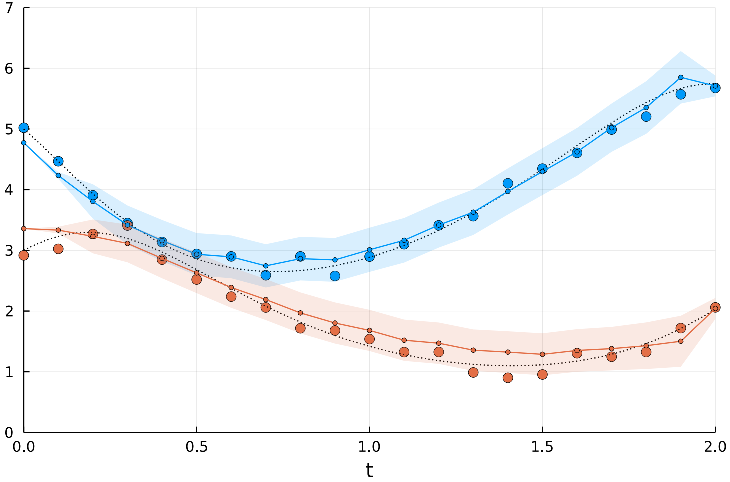


Figure: i=66

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

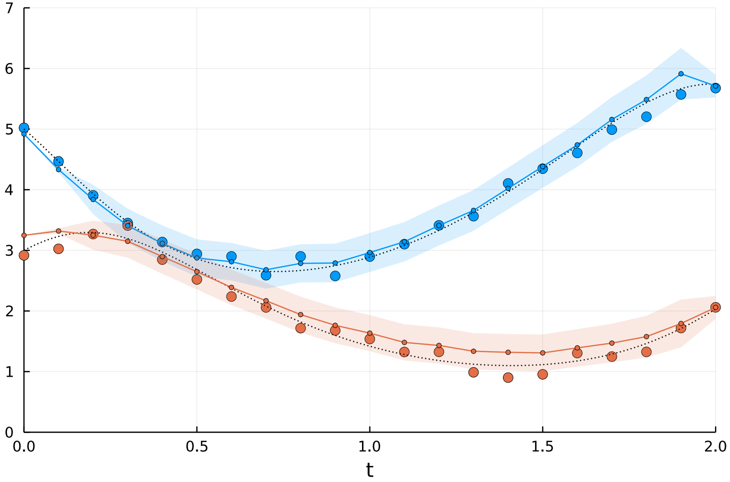


Figure: i=67

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

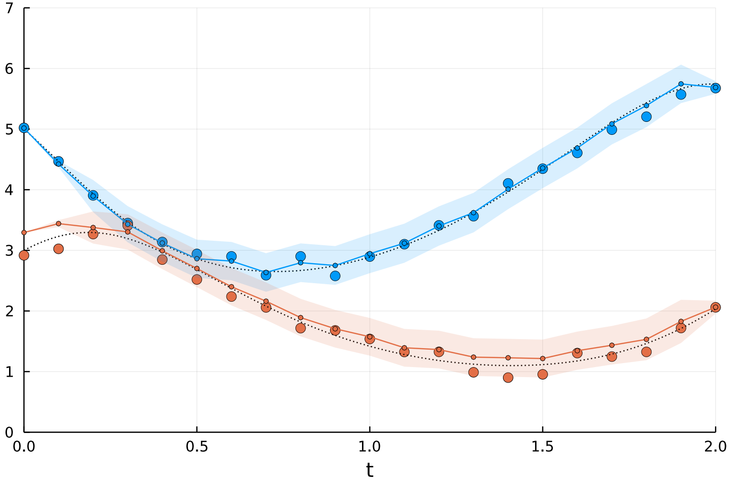


Figure: i=68

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

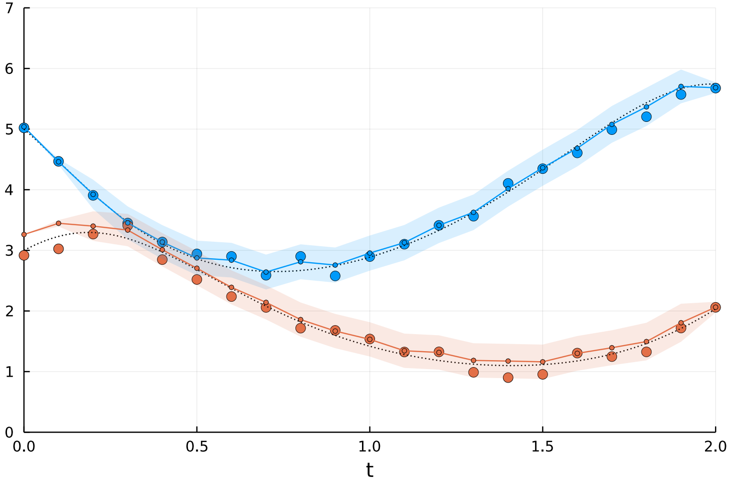


Figure: i=69

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

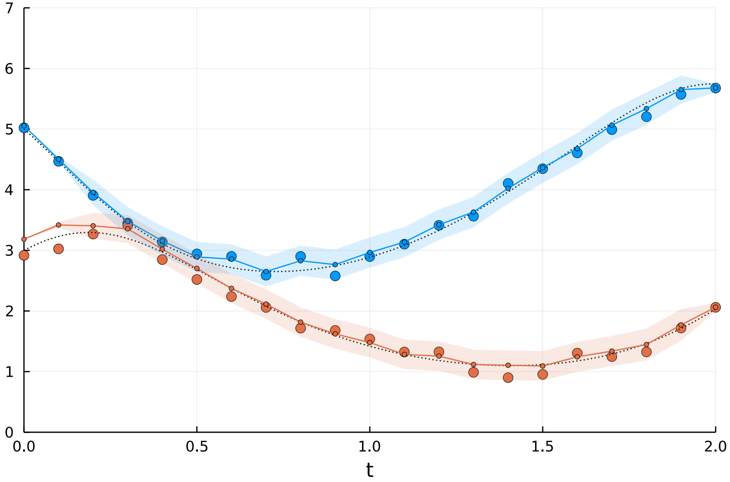


Figure: i=70

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

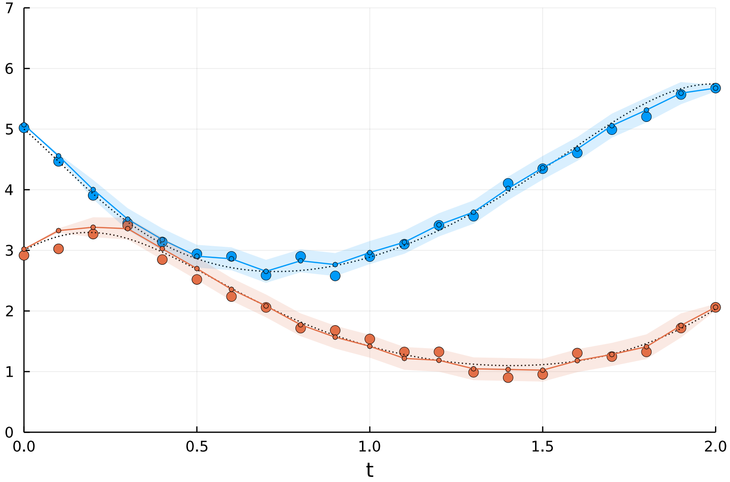


Figure: i=71

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

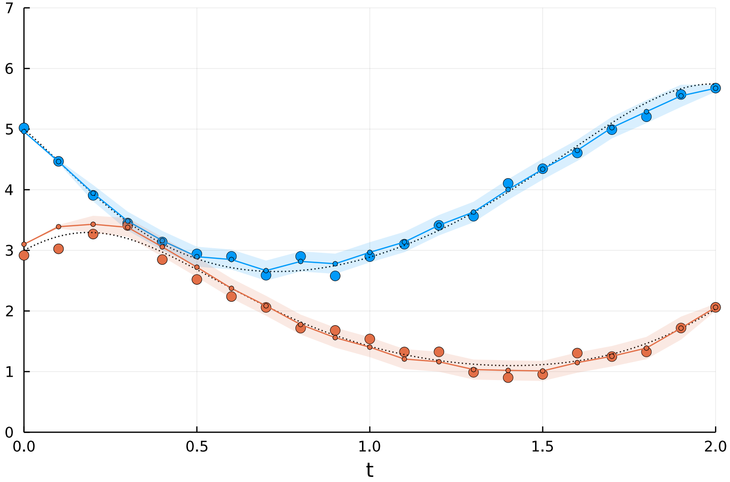


Figure: i=72

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

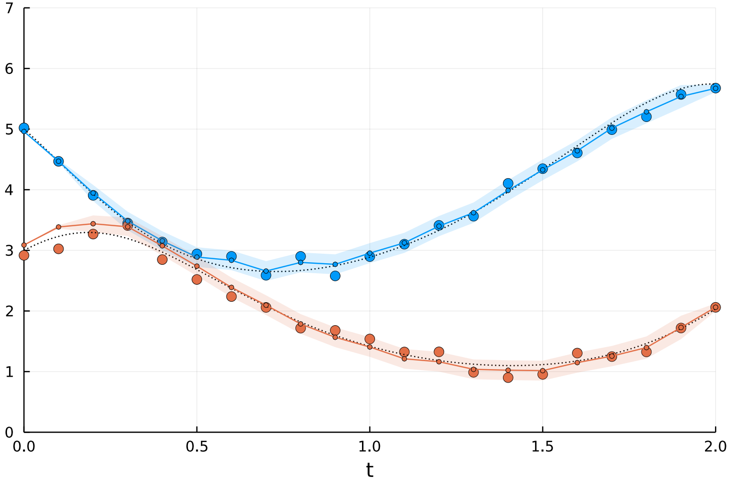


Figure: i=73

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

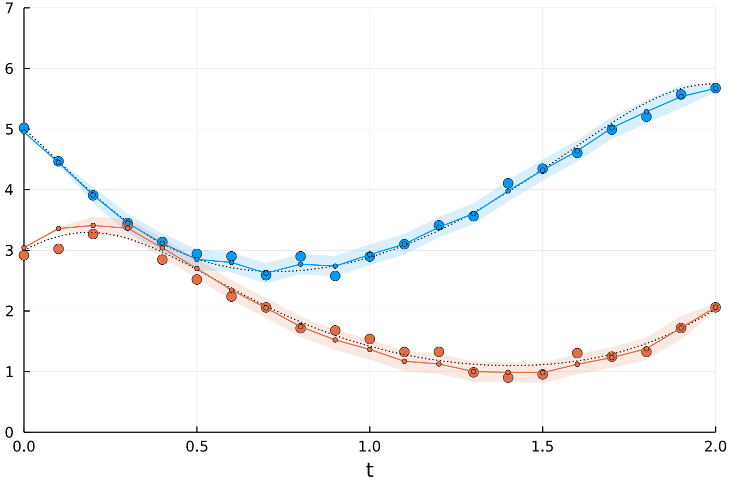


Figure: i=74

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

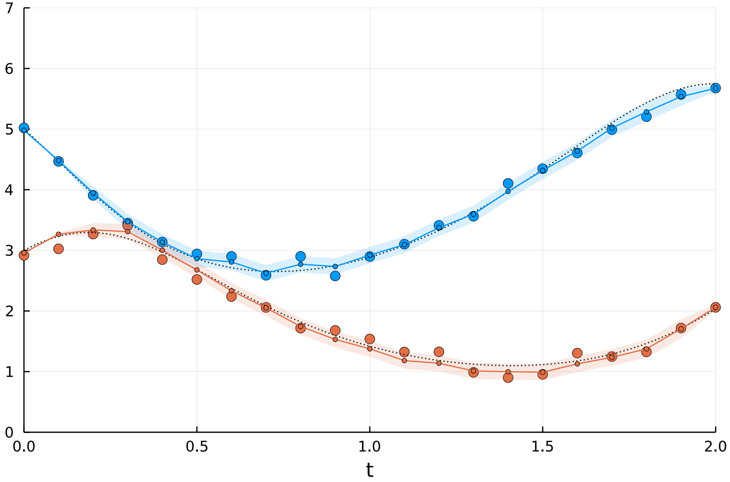


Figure: i=75

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

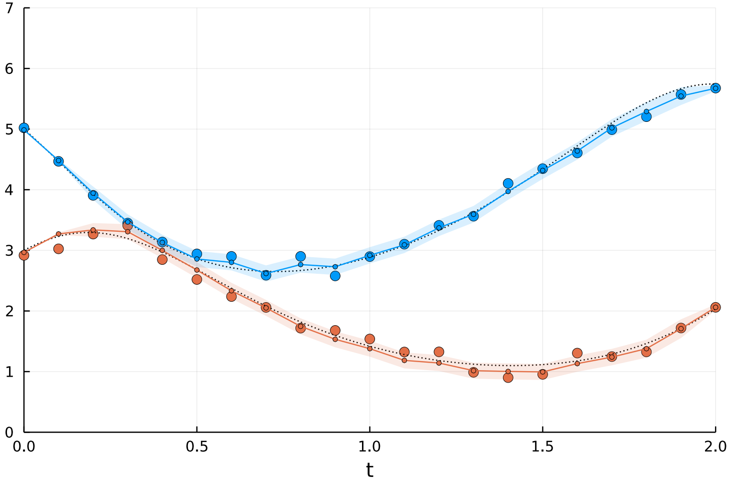


Figure: i=76

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

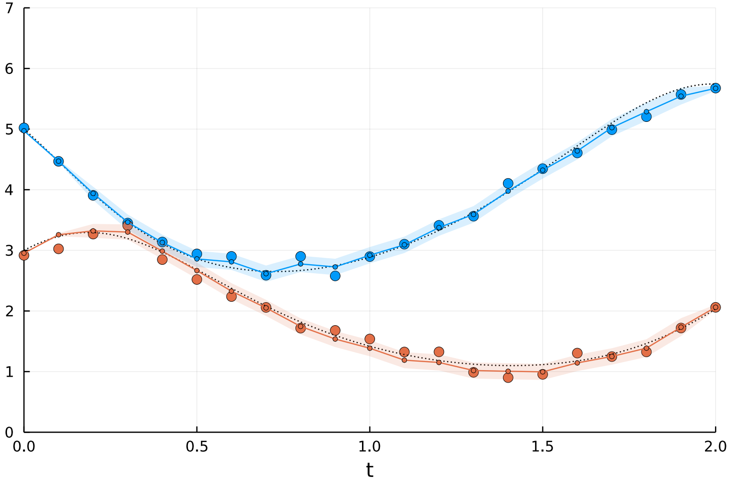


Figure: i=77

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

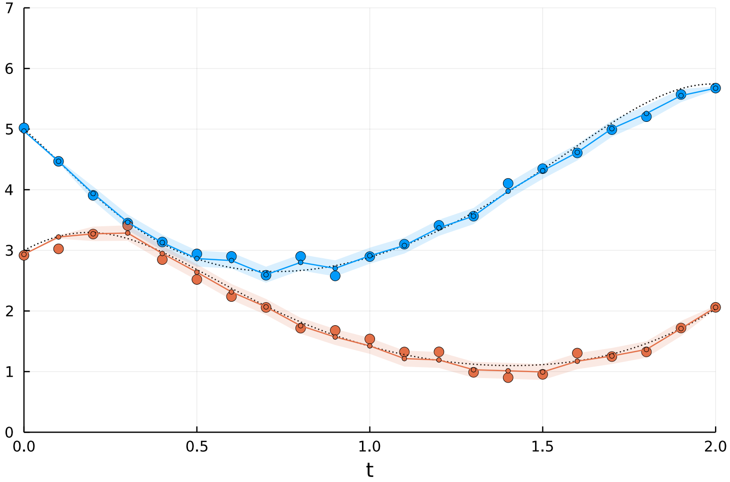


Figure: i=78

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

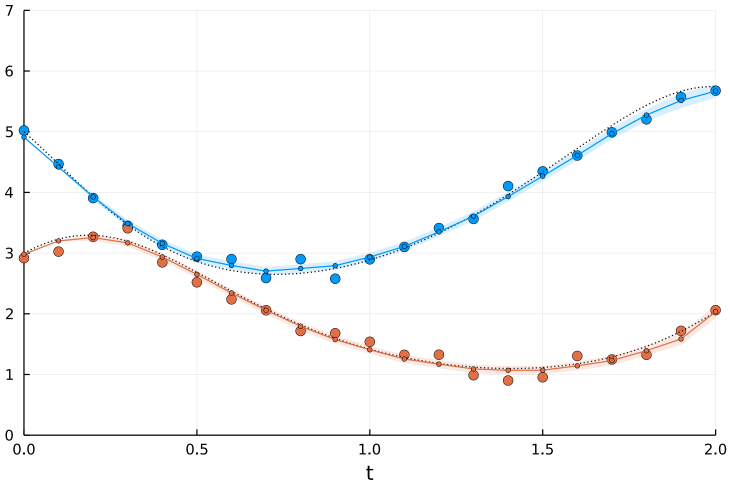


Figure: i=79

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

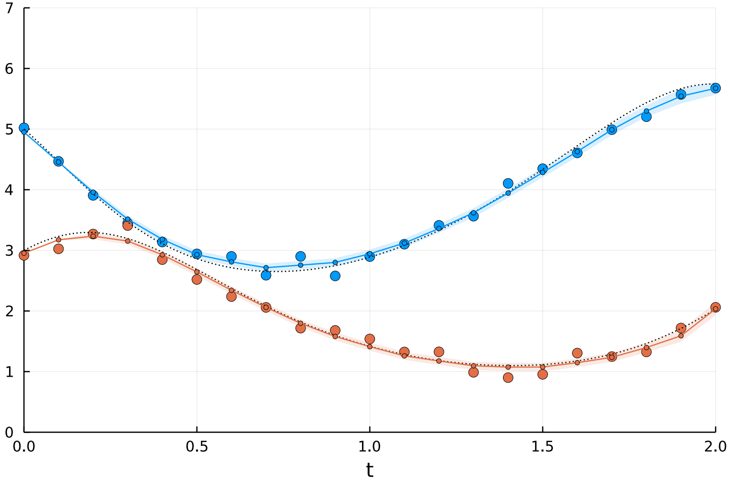


Figure: i=80

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

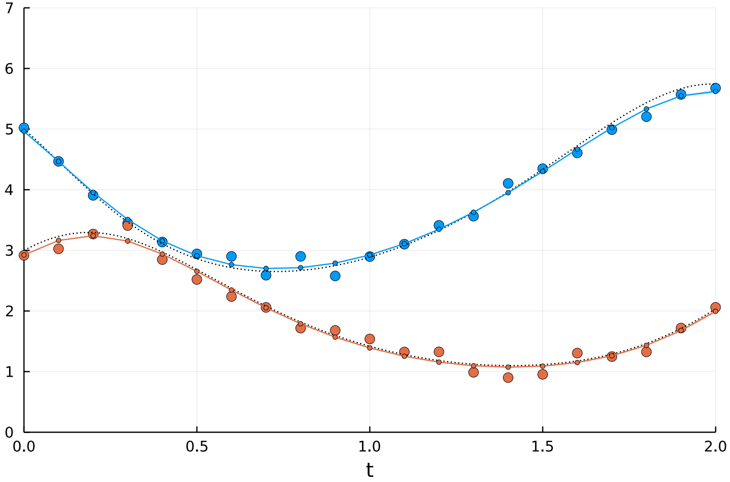


Figure: i=90

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

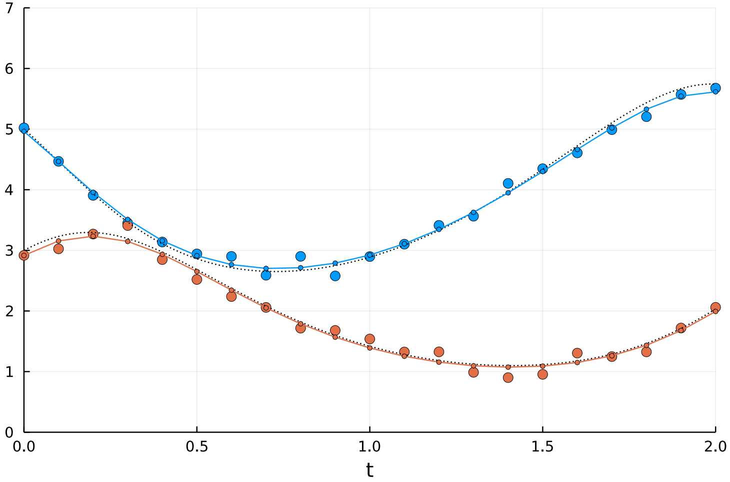


Figure: i=100

Example: Probabilistic Numerical Integration

Optimizing ODE parameters and prior hyperparameters jointly

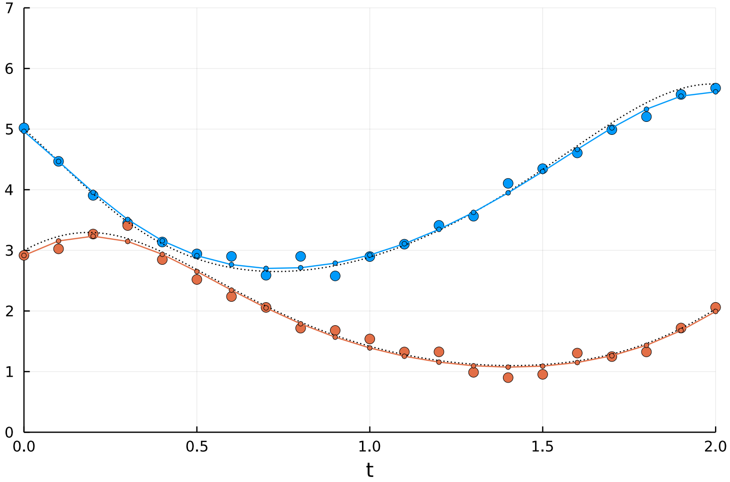
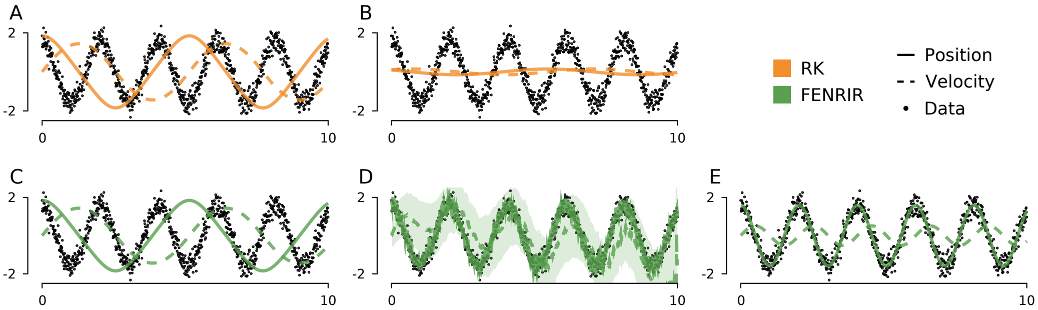


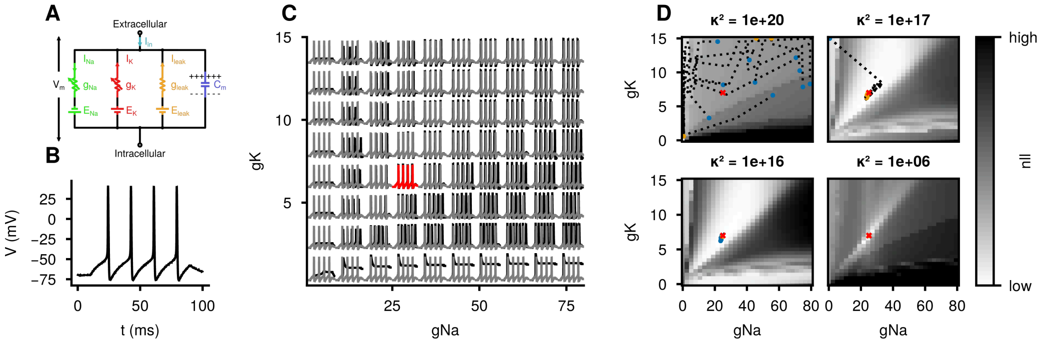
Figure: i=100 DONE

Inference in a partially observed oscillatory system

The probabilistic solver can escape the local optimum



Gradient-based parameter inference in a Hodgkin–Huxley neuron



Summary

- *ODE solving is state estimation*
  $\Rightarrow$ treat initial value problems as state estimation problems
- *"ODE filters"*: **How to solve ODEs with Bayesian filtering and smoothing**
- *Flexible information operators* to solve more than just standard ODEs
- *Parameter inference:* Being uncertain about the ODE solution allows you to update on data

Software packages

```
https://github.com/nathanaelbosch/ProbNumDiffEq.jl
]add ProbNumDiffEq
```

```
https://github.com/probabilistic-numerics/probnum
pip install probnum
```

```
https://github.com/pnkraemer/probdiffeq
pip install probdiffeq
```

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Bosch, N., Corenflos, A., Yaghoobi, F., Tronarp, F., Hennig, P., and Särkkä, S. (2023a).
  Parallel-in-time probabilistic numerical ODE solvers.
- Bosch, N., Hennig, P., and Tronarp, F. (2021).
  Calibrated adaptive probabilistic ODE solvers.
  In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3466–3474. PMLR.
- Bosch, N., Hennig, P., and Tronarp, F. (2023b).
  Probabilistic exponential integrators.
  In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Bosch, N., Tronarp, F., and Hennig, P. (2022).
  Pick-and-mix information operators for probabilistic ODE solvers.
  In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 10015–10027. PMLR.

Bibliography II

- Kersting, H., Sullivan, T. J., and Hennig, P. (2020).
  Convergence rates of gaussian ode filters.
  Statistics and Computing, 30(6):1791–1816.
- Krämer, N., Bosch, N., Schmidt, J., and Hennig, P. (2022).
  Probabilistic ODE solutions in millions of dimensions.
  In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11634–11649. PMLR.
- Krämer, N. and Hennig, P. (2020).
  Stable implementation of probabilistic ode solvers.
  CoRR.
- Krämer, N. and Hennig, P. (2021).
  Linear-time probabilistic solution of boundary value problems.
  In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11160–11171. Curran Associates, Inc.

Bibliography III

- Krämer, N., Schmidt, J., and Hennig, P. (2022).
  Probabilistic numerical method of lines for time-dependent partial differential equations.
  In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 625–639. PMLR.
- Schmidt, J., Krämer, N., and Hennig, P. (2021).
  A probabilistic state space model for joint inference from differential equations and data.
  In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12374–12385. Curran Associates, Inc.
- Tronarp, F., Bosch, N., and Hennig, P. (2022).
  Fenrir: Physics-enhanced regression for initial value problems.
  In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21776–21794. PMLR.

Bibliography IV

- Tronarp, F., Kersting, H., Särkkä, S., and Hennig, P. (2019).
  Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective.
  Statistics and Computing, 29(6):1297–1315.
- Tronarp, F., Särkkä, S., and Hennig, P. (2021).
  Bayesian ode solvers: the maximum a posteriori estimate.
  Statistics and Computing, 31(3):23.