Numerics of Machine Learning
Lecture 07
Probabilistic Numerical ODE Solvers

Nathanael Bosch & Jonathan Schmidt

1 December 2022

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

Two weeks ago: **State-space models and extended Kalman filters/smoothers**

- "How to estimate the *state* of a dynamical system from *observations*"

Last week: **Ordinary differential equations and how to solve them**

- "How to *simulate*, approximately, a deterministic dynamical system"

This week: **ODE simulation as probabilistic inference**

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

Two weeks ago: **State-space models and extended Kalman filters/smoothers**

- "How to estimate the *state* of a dynamical system from *observations*"

Last week: **Ordinary differential equations and how to solve them**

- "How to *simulate*, approximately, a deterministic dynamical system"

This week: **ODE simulation as probabilistic inference**

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

  $\Rightarrow$ ***Probabilistic* numerical ODE solvers**

Recap: Numerical Ordinary Differential Equation Solvers

Recap: Ordinary Differential Equations and how to solve them

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

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

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $x(0) = x_0$. Goal: "Find x".

Recap: Ordinary Differential Equations and how to solve them

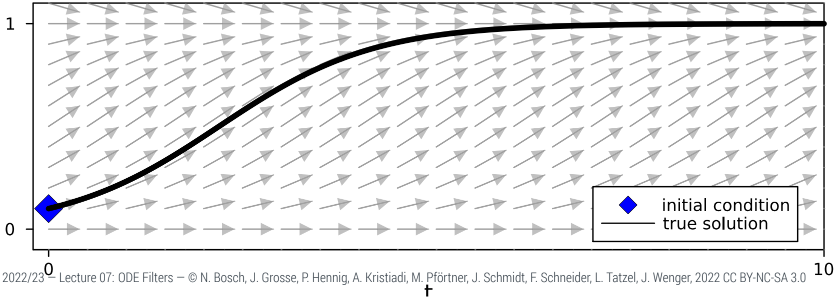
Numerical ODE solvers try to estimate an unknown function by evaluating the vector field

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

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $x(0) = x_0$. Goal: "Find x".

- **Simple example**: Logistic ODE

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



Recap: Ordinary Differential Equations and how to solve them

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

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

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $x(0) = x_0$. Goal: "Find x".

Numerical ODE solvers:

- Forward Euler:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot f(\hat{x}(t), t)$$

- Backward Euler:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot f(\hat{x}(t+h), t+h)$$

- Runge–Kutta:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot \sum_{i=1}^{s} b_i f(\tilde{x}_i, t + c_i h)$$

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

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $x(0) = x_0$. Goal: "Find x".

Numerical ODE solvers:

- Forward Euler:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot f(\hat{x}(t), t)$$

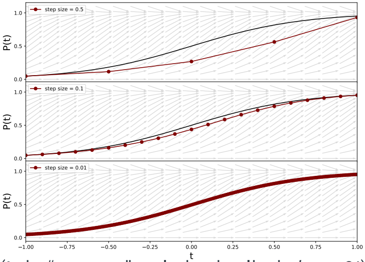
- Backward Euler:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot f(\hat{x}(t+h), t+h)$$

- Runge–Kutta:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot \sum_{i=1}^{s} b_i f(\tilde{x}_i, t + c_i h)$$

Forward Euler for different step sizes:



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

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

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

with $t \in [0, T]$, vector field $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, and initial value $x(0) = x_0$. Goal: "Find x".

Numerical ODE solvers:

- Forward Euler:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot f(\hat{x}(t), t)$$

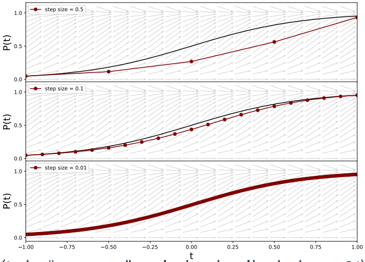
- Backward Euler:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot f(\hat{x}(t+h), t+h)$$

- Runge–Kutta:

$$\hat{x}(t+h) = \hat{x}(t) + h \cdot \sum_{i=1}^{s} b_i f(\tilde{x}_i, t + c_i h)$$

Forward Euler for different step sizes:



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

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

Recap: **Bayesian State Estimation with Extended Kalman filtering and smoothing**

Recap: Extended Kalman filtering and smoothing

EKF/EKS as introduced in lecture 5

Non-linear Gaussian state-estimation problem:

$$
\begin{aligned}
\text{Initial distribution:} \quad & x_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \\
\text{Prior / dynamics model:} \quad & x_{i+1} \mid x_i \sim \mathcal{N}(f(x_i), Q_i), \\
\text{Likelihood / measurement model:} \quad & y_i \mid x_i \sim \mathcal{N}(h(x_i), R_i), \\
\text{Data:} \quad & \mathcal{D} = \{y_i\}_{i=1}^N.
\end{aligned}
$$

Recap: Extended Kalman filtering and smoothing

EKF/EKS as introduced in lecture 5

Non-linear Gaussian state-estimation problem:

$$
\begin{aligned}
\text{Initial distribution:} \quad & x_0 \sim \mathcal{N}\left(\mu_0, \Sigma_0\right), \\
\text{Prior / dynamics model:} \quad & x_{i+1} \mid x_i \sim \mathcal{N}\left(f(x_i), Q_i\right), \\
\text{Likelihood / measurement model:} \quad & y_i \mid x_i \sim \mathcal{N}\left(h(x_i), R_i\right), \\
\text{Data:} \quad & \mathcal{D} = \{y_i\}_{i=1}^N .
\end{aligned}
$$

EKF/EKS: The extended Kalman filter/smoother recursively computes Gaussian approximations:

$$
\begin{aligned}
\text{Predict:} \quad & p(x_i \mid y_{1:i-1}) \approx \mathcal{N}(x_i; \mu_i^P, \Sigma_i^P), \\
\text{Filter:} \quad & p(x_i \mid y_{1:i}) \approx \mathcal{N}(x_i; \mu_i, \Sigma_i), \\
\text{Smooth:} \quad & p(x_i \mid y_{1:N}) \approx \mathcal{N}(x_i; \mu_i^S, \Sigma_i^S), \\
\text{Likelihood:} \quad & p(y_i \mid y_{1:i-1}) \approx \mathcal{N}(y_i; \hat{y}_i, S_i).
\end{aligned}
$$

Recap: Extended Kalman filtering and smoothing

EKF/EKS as introduced in lecture 5

Non-linear Gaussian state-estimation problem:

$$
\begin{aligned}
\text{Initial distribution:} \quad & x_0 \sim \mathcal{N}\left(\mu_0, \Sigma_0\right), \\
\text{Prior / dynamics model:} \quad & x_{i+1} \mid x_i \sim \mathcal{N}\left(f(x_i), Q_i\right), \\
\text{Likelihood / measurement model:} \quad & y_i \mid x_i \sim \mathcal{N}\left(h(x_i), R_i\right), \\
\text{Data:} \quad & \mathcal{D} = \{y_i\}_{i=1}^N.
\end{aligned}
$$

EKF/EKS: The extended Kalman filter/smoother recursively computes Gaussian approximations:

$$
\begin{aligned}
\text{Predict:} \quad & p(x_i \mid y_{1:i-1}) \approx \mathcal{N}(x_i; \mu_i^P, \Sigma_i^P), \\
\text{Filter:} \quad & p(x_i \mid y_{1:i}) \approx \mathcal{N}(x_i; \mu_i, \Sigma_i), \\
\text{Smooth:} \quad & p(x_i \mid y_{1:N}) \approx \mathcal{N}(x_i; \mu_i^S, \Sigma_i^S), \\
\text{Likelihood:} \quad & p(y_i \mid y_{1:i-1}) \approx \mathcal{N}(y_i; \hat{y}_i, S_i).
\end{aligned}
$$

PREDICT

$$
\begin{aligned}
\mu_{i+1}^P &= f(\mu_i), \\
\Sigma_{i+1}^P &= J_f(\mu_i)\Sigma_i J_f(\mu_i)^\top + Q_i.
\end{aligned}
$$

Recap: Extended Kalman filtering and smoothing

EKF/EKS as introduced in lecture 5

Non-linear Gaussian state-estimation problem:

$$
\begin{aligned}
\text{Initial distribution:} \quad & x_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \\
\text{Prior / dynamics model:} \quad & x_{i+1} \mid x_i \sim \mathcal{N}(f(x_i), Q_i), \\
\text{Likelihood / measurement model:} \quad & y_i \mid x_i \sim \mathcal{N}(h(x_i), R_i), \\
\text{Data:} \quad & \mathcal{D} = \{y_i\}_{i=1}^N.
\end{aligned}
$$

EKF/EKS: The extended Kalman filter/smoother recursively computes Gaussian approximations:

$$
\begin{aligned}
\text{Predict:} \quad & p(x_i \mid y_{1:i-1}) \approx \mathcal{N}(x_i; \mu_i^P, \Sigma_i^P), \\
\text{Filter:} \quad & p(x_i \mid y_{1:i}) \approx \mathcal{N}(x_i; \mu_i, \Sigma_i), \\
\text{Smooth:} \quad & p(x_i \mid y_{1:N}) \approx \mathcal{N}(x_i; \mu_i^S, \Sigma_i^S), \\
\text{Likelihood:} \quad & p(y_i \mid y_{1:i-1}) \approx \mathcal{N}(y_i; \hat{y}_i, S_i).
\end{aligned}
$$

PREDICT

$$
\begin{aligned}
\mu_{i+1}^P &= f(\mu_i), \\
\Sigma_{i+1}^P &= J_f(\mu_i)\Sigma_i J_f(\mu_i)^\top + Q_i.
\end{aligned}
$$

UPDATE

$$
\begin{aligned}
\hat{z}_i &= h(\mu_i^P), \\
S_i &= J_h(\mu_i^P)\Sigma_i^P J_h(\mu_i^P)^\top + R_i, \\
K_i &= \Sigma_i^P J_h(\mu_i^P)^\top S_i^{-1}, \\
\mu_i &= \mu_i^P + K_i (z_i - \hat{z}_i), \\
\Sigma_i &= \Sigma_i^P - K_i S_i K_i^\top.
\end{aligned}
$$

Recap: Extended Kalman filtering and smoothing

EKF/EKS as introduced in lecture 5

Non-linear Gaussian state-estimation problem:

$$
\begin{aligned}
\text{Initial distribution:} \quad & x_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \\
\text{Prior / dynamics model:} \quad & x_{i+1} \mid x_i \sim \mathcal{N}(f(x_i), Q_i), \\
\text{Likelihood / measurement model:} \quad & y_i \mid x_i \sim \mathcal{N}(h(x_i), R_i), \\
\text{Data:} \quad & \mathcal{D} = \{y_i\}_{i=1}^N.
\end{aligned}
$$

EKF/EKS: The extended Kalman filter/smoother recursively computes Gaussian approximations:

$$
\begin{aligned}
\text{Predict:} \quad & p(x_i \mid y_{1:i-1}) \approx \mathcal{N}(x_i; \mu_i^P, \Sigma_i^P), \\
\text{Filter:} \quad & p(x_i \mid y_{1:i}) \approx \mathcal{N}(x_i; \mu_i, \Sigma_i), \\
\text{Smooth:} \quad & p(x_i \mid y_{1:N}) \approx \mathcal{N}(x_i; \mu_i^S, \Sigma_i^S), \\
\text{Likelihood:} \quad & p(y_i \mid y_{1:i-1}) \approx \mathcal{N}(y_i; \hat{y}_i, S_i).
\end{aligned}
$$

PREDICT

$$
\begin{aligned}
\mu_{i+1}^P &= f(\mu_i), \\
\Sigma_{i+1}^P &= J_f(\mu_i)\Sigma_i J_f(\mu_i)^\top + Q_i.
\end{aligned}
$$

UPDATE

$$
\begin{aligned}
\hat{z}_i &= h(\mu_i^P), \\
S_i &= J_h(\mu_i^P)\Sigma_i^P J_h(\mu_i^P)^\top + R_i, \\
K_i &= \Sigma_i^P J_h(\mu_i^P)^\top S_i^{-1}, \\
\mu_i &= \mu_i^P + K_i (z_i - \hat{z}_i), \\
\Sigma_i &= \Sigma_i^P - K_i S_i K_i^\top.
\end{aligned}
$$

SMOOTH: See lecture 5.

Today: *Probabilistic* **numerical ODE solutions**

or “how to treat ODEs as the state estimation problem that they really are”

Probabilistic numerical ODE solutions

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

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference
$\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference
$\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

$\Rightarrow$ We need to construct a state-space model:

1. Prior:
2. Likelihood:
3. Data:

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference
$\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

$\Rightarrow$ We need to construct a state-space model:

1. **Prior:**
2. Likelihood:
3. Data:

The Prior: Describing the evolution of the "state"

What is the "state", and how can we model it?

The Prior: Describing the evolution of the “state”

What is the “state”, and how can we model it?

- **What is the “state”?** Only $x(t)$ is not sufficient to fully describe the dynamical system.

The Prior: Describing the evolution of the "state"

What is the "state", and how can we model it?

- **What is the "state"?** Only $x(t)$ is not sufficient to fully describe the dynamical system.
- **Motivation** from Taylor series expansions:

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + \cdots = \sum_{k=0}^{\infty} \frac{h^k}{k!} x^{(k)}(t).$$

$\Rightarrow$ Having access to *all* derivatives *would* fully describe the dynamical system.

The Prior: Describing the evolution of the "state"

What is the "state", and how can we model it?

- **What is the "state"?** Only $x(t)$ is not sufficient to fully describe the dynamical system.
- **Motivation** from Taylor series expansions:

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + \cdots = \sum_{k=0}^{\infty} \frac{h^k}{k!} x^{(k)}(t).$$

  $\Rightarrow$ Having access to *all* derivatives *would* fully describe the dynamical system.
- **Truncating the Taylor series expansion** greatly simplifies things, but introduces an error term:

$$x(t+h) = \sum_{k=0}^{q} \frac{h^k}{k!} x^{(k)}(t) + \mathcal{O}(h^{q+1}).$$

The Prior: Describing the evolution of the “state”

What is the “state”, and how can we model it?

- **What is the “state”?** Only $x(t)$ is not sufficient to fully describe the dynamical system.
- **Motivation** from Taylor series expansions:

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + \cdots = \sum_{k=0}^{\infty} \frac{h^k}{k!} x^{(k)}(t).$$

$\Rightarrow$ Having access to *all* derivatives *would* fully describe the dynamical system.

- **Truncating the Taylor series expansion** greatly simplifies things, but introduces an error term:

$$x(t+h) = \sum_{k=0}^{q} \frac{h^k}{k!} x^{(k)}(t) + \mathcal{O}(h^{q+1}).$$

- **Towards a “prior process”:** Modeling the error with a random variable gives

$$x(t+h) = \sum_{k=0}^{q} \frac{h^k}{k!} x^{(k)}(t) + \epsilon(t,h), \qquad \epsilon(t,h) \sim \mathcal{N}\left(0, \frac{h^{2q+1}}{(2q+1)\,(q!)^2}\right)$$

The Prior: Describing the evolution of the "state"

What is the "state", and how can we model it?

- **What is the "state"?** Only $x(t)$ is not sufficient to fully describe the dynamical system.
- **Motivation** from Taylor series expansions:

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + \cdots = \sum_{k=0}^{\infty} \frac{h^k}{k!} x^{(k)}(t).$$

  $\Rightarrow$ Having access to *all* derivatives *would* fully describe the dynamical system.
- **Truncating the Taylor series expansion** greatly simplifies things, but introduces an error term:

$$x(t+h) = \sum_{k=0}^{q} \frac{h^k}{k!} x^{(k)}(t) + \mathcal{O}(h^{q+1}).$$

- **Towards a "prior process":** Modeling the error with a random variable gives

$$x(t+h) = \sum_{k=0}^{q} \frac{h^k}{k!} x^{(k)}(t) + \epsilon(t,h), \qquad \epsilon(t,h) \sim \mathcal{N}\left(0, \frac{h^{2q+1}}{(2q+1)\,(q!)^2}\right)$$

- **This motivates a "state"**: $X(t) = \left[x(t), \dot{x}(t), \ddot{x}(t), \ldots, x^{(q)}(t)\right]^\top$. *Details on the next slide.*

The Prior: Describing the evolution of the "state"

The q-times integrated Wiener process

The prior model: *q-times integrated Wiener process* (IWP(q)).

The Prior: Describing the evolution of the "state"

The q-times integrated Wiener process

The prior model: *q-times integrated Wiener process* ($\mathrm{IWP}(q)$).

Let $X(t)$ be of the form

$$X(t) = \left[X^{(0)}(t), X^{(1)}(t), \dots, X^{(q)}(t)\right]^T,$$

chosen such that $X^{(i)}(t)$ models the i-th derivative of $x(t)$.

The prior model: *q-times integrated Wiener process* (IWP(q)).

Let $X(t)$ be of the form

$$X(t) = \left[X^{(0)}(t), X^{(1)}(t), \dots, X^{(q)}(t)\right]^T,$$

chosen such that $\underline{X^{(i)}(t) \text{ models the } i\text{-th derivative of } x(t)}$.

The IWP(q) has known discrete-time transitions:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), \sigma^2 Q(h)\right),$$

$$[A(h)]_{ij} = \mathbb{1}_{i \leq j} \frac{h^{j-i}}{(j-i)!},$$

$$[Q(h)]_{ij} = \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!},$$

for any $i, j = 0, \dots, q$.

The prior model: *q-times integrated Wiener process* (IWP(q)).

Let $X(t)$ be of the form

$$X(t) = \left[X^{(0)}(t), X^{(1)}(t), \dots, X^{(q)}(t)\right]^T,$$

chosen such that $X^{(i)}(t)$ models the i-th derivative of $x(t)$.

The IWP(q) has known discrete-time transitions:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), \sigma^2 Q(h)\right),$$

$$[A(h)]_{ij} = \mathbb{1}_{i \leq j} \frac{h^{j-i}}{(j-i)!},$$

$$[Q(h)]_{ij} = \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!},$$

for any $i, j = 0, \dots, q$.

Example: IWP(2)

$$A(h) = \begin{pmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{pmatrix},$$

$$Q(h) = \begin{pmatrix} \frac{h^5}{20} & \frac{h^4}{8} & \frac{h^3}{6} \\ \frac{h^4}{8} & \frac{h^3}{3} & \frac{h^2}{2} \\ \frac{h^3}{6} & \frac{h^2}{2} & h \end{pmatrix}.$$

The prior model: *q-times integrated Wiener process* (IWP(q)).

Let $X(t)$ be of the form

$$X(t) = \left[X^{(0)}(t), X^{(1)}(t), \dots, X^{(q)}(t)\right]^T,$$

chosen such that $\underline{X^{(i)}(t) \text{ models the } i\text{-th derivative of } x(t)}$.

The IWP(q) has known discrete-time transitions:

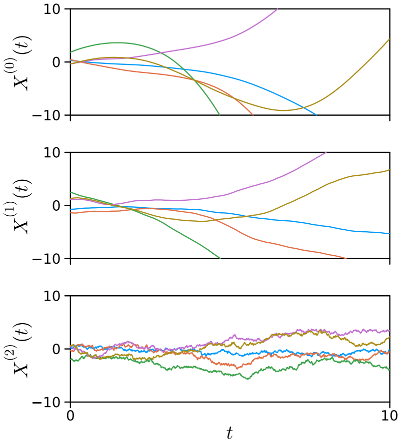
$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), \sigma^2 Q(h)\right),$$

$$[A(h)]_{ij} = \mathbb{1}_{i \leq j} \frac{h^{j-i}}{(j-i)!},$$

$$[Q(h)]_{ij} = \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!},$$

for any $i, j = 0, \dots, q$.

Example: IWP(2)



The prior model: *q-times integrated Wiener process* (IWP(q)).

Let $X(t)$ be of the form

$$X(t) = \left[X^{(0)}(t), X^{(1)}(t), \dots, X^{(q)}(t)\right]^T,$$

chosen such that $X^{(i)}(t)$ models the i-th derivative of $x(t)$.

The IWP(q) has known discrete-time transitions:

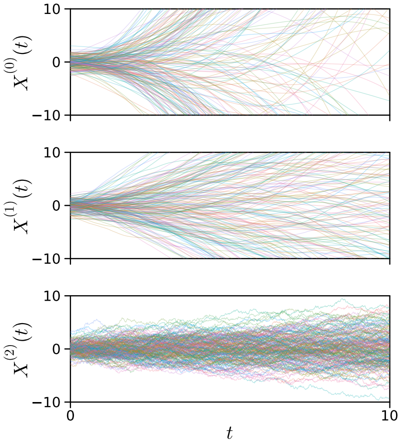
$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), \sigma^2 Q(h)\right),$$

$$[A(h)]_{ij} = \mathbb{1}_{i \leq j} \frac{h^{j-i}}{(j-i)!},$$

$$[Q(h)]_{ij} = \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!},$$

for any $i, j = 0, \dots, q$.

Example: IWP(2)



The prior model: *q-times integrated Wiener process* (IWP(q)).

Let $X(t)$ be of the form

$$X(t) = \left[X^{(0)}(t), X^{(1)}(t), \dots, X^{(q)}(t)\right]^T,$$

chosen such that $X^{(i)}(t)$ models the i-th derivative of $x(t)$.

The IWP(q) has known discrete-time transitions:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), \sigma^2 Q(h)\right),$$

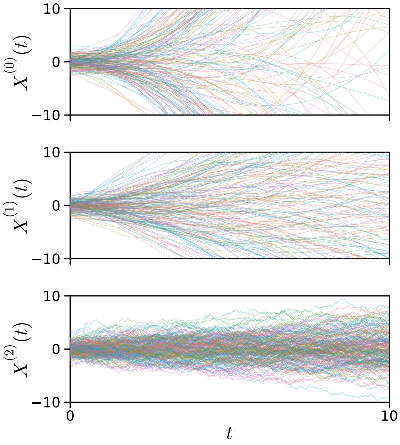
$$[A(h)]_{ij} = \mathbb{1}_{i \leq j} \frac{h^{j-i}}{(j-i)!},$$

$$[Q(h)]_{ij} = \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!},$$

for any $i, j = 0, \dots, q$.

For later convenience: Define projection matrices $E_i X = X^{(i)}$.

Example: IWP(2)



Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference $\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

1. **Prior:** q-times integrated Wiener process prior:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), Q(h)\right)$$

2. Likelihood:
3. Data:

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference $\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

1. Prior: q-times integrated Wiener process prior:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(X(t+h); A(h)X(t), Q(h)\right)$$

2. **Likelihood:**
3. **Data:**

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

The likelihood model and the data — aka. “*The information operator*”

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), t) \\
\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) &= f\left(X^{(0)}(t), t\right) \\
0 &= X^{(1)}(t) - f\left(X^{(0)}(t), t\right)
\end{aligned}
$$

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \quad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \quad i = 1, \dots, N.$$

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \ldots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

The likelihood model and the data — aka. "*The information operator*"

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \quad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^{N}$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \dots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

- This motivates the **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \mathcal{N}\left(m(X(t_i), t_i), R\right)$$

$$z_i \triangleq 0, \qquad i = 1, \dots, N.$$

where z_i is a realization of $Z(t_i)$.

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), t) \\
\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) &= f\left(X^{(0)}(t), t\right) \\
0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) &=: m(X(t), t).
\end{aligned}
$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^{N}$

$$
\begin{aligned}
\dot{x}(t_i) &= f(x(t_i), t_i), \qquad i = 1, \dots, N. \\
\Leftrightarrow \qquad m(X(t_i), t_i) &= 0
\end{aligned}
$$

- This motivates the *noiseless* **likelihood model** and **data**:

$$
\begin{aligned}
Z(t_i) \mid X(t_i) &\sim \mathcal{N}\left(m(X(t_i), t_i), 0\right) \\
z_i &\triangleq 0, \qquad i = 1, \dots, N.
\end{aligned}
$$

where z_i is a realization of $Z(t_i)$.

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \ldots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

- This motivates the *noiseless* **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \delta\left(m(X(t_i), t_i)\right)$$

$$z_i \triangleq 0, \qquad i = 1, \ldots, N.$$

where z_i is a realization of $Z(t_i)$.
(δ is the Dirac distribution)

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \ldots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

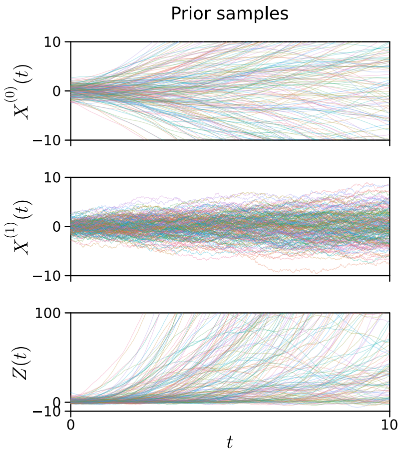
- This motivates the *noiseless* **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \delta\left(m(X(t_i), t_i)\right)$$

$$z_i \triangleq 0, \qquad i = 1, \ldots, N.$$

where z_i is a realization of $Z(t_i)$.
(δ is the Dirac distribution)

Example: Logistic ODE $\dot{x} = x(1 - x)$



(here: $Z = X^{(1)} - X^{(0)}(1 - X^{(0)})$)

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \ldots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

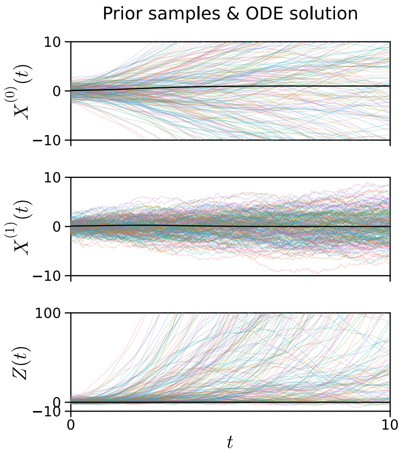
- This motivates the *noiseless* **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \delta\left(m(X(t_i), t_i)\right)$$

$$z_i \triangleq 0, \qquad i = 1, \ldots, N.$$

where z_i is a realization of $Z(t_i)$.
(δ is the Dirac distribution)

Example: Logistic ODE $\dot{x} = x(1-x)$



(here: $Z = X^{(1)} - X^{(0)}(1 - X^{(0)})$)

The likelihood model and the data — aka. "*The information operator*"

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \quad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) \quad =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^{N}$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \dots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

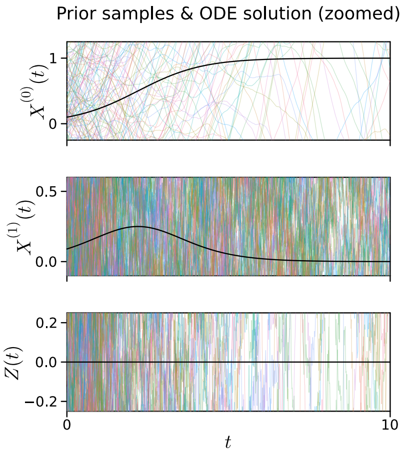
- This motivates the *noiseless* **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \delta\left(m(X(t_i), t_i)\right)$$

$$z_i \triangleq 0, \qquad i = 1, \dots, N.$$

where z_i is a realization of $Z(t_i)$.
(δ is the Dirac distribution)

Example: Logistic ODE $\dot{x} = x(1 - x)$



(here: $Z = X^{(1)} - X^{(0)}(1 - X^{(0)})$)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \qquad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) \;=:\; m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \qquad i = 1, \ldots, N.$$

$$\Leftrightarrow \qquad m(X(t_i), t_i) = 0$$

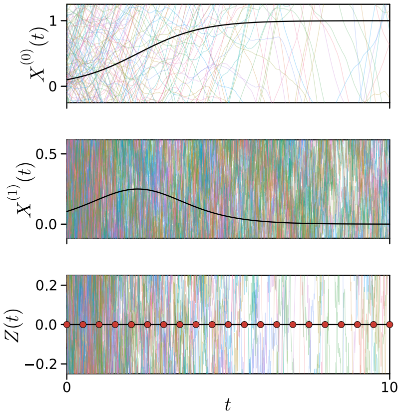
- This motivates the *noiseless* **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \delta\left(m(X(t_i), t_i)\right)$$

$$z_i \triangleq 0, \qquad i = 1, \ldots, N.$$

where z_i is a realization of $Z(t_i)$.
(δ is the Dirac distribution)

Example: Logistic ODE $\dot{x} = x(1 - x)$



(here: $Z = X^{(1)} - X^{(0)}(1 - X^{(0)})$)

The likelihood model and the data — aka. "*The information operator*"

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

The likelihood and data relate the prior to the desired posterior: the numerical ODE solution

- **Ideal but intractable goal:** Want $x(t)$ to satisfy the ODE

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

$$\overset{\text{using } X(t)}{\Leftrightarrow} \quad X^{(1)}(t) = f\left(X^{(0)}(t), t\right)$$

$$0 = X^{(1)}(t) - f\left(X^{(0)}(t), t\right) =: m(X(t), t).$$

- **Easier goal:** Satisfy the ODE *on a discrete time grid* $\{t_i\}_{i=1}^N$

$$\dot{x}(t_i) = f(x(t_i), t_i), \quad i = 1, \dots, N.$$

$$\Leftrightarrow \quad m(X(t_i), t_i) = 0$$

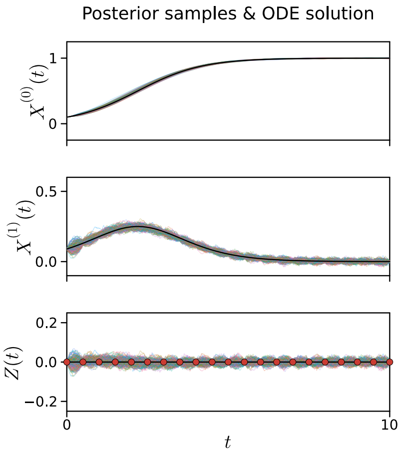
- This motivates the *noiseless* **likelihood model** and **data**:

$$Z(t_i) \mid X(t_i) \sim \delta\left(m(X(t_i), t_i)\right)$$

$$z_i \triangleq 0, \qquad i = 1, \dots, N.$$

where z_i is a realization of $Z(t_i)$.
(δ is the Dirac distribution)

Example: Logistic ODE $\dot{x} = x(1 - x)$



(here: $Z = X^{(1)} - X^{(0)}(1 - X^{(0)})$)

Spoiler: **This is the thing we want!**

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference $\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

1. Prior: q-times integrated Wiener process prior:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), Q(h)\right)$$

2. **Likelihood:** $Z(t) \mid X(t) \sim \delta\left(X^{(1)}(t) - f(X^{(0)}(t), t)\right)$
3. **Data:** $\mathcal{D}_{\mathrm{PN}} = \{z_i\}$, with $(Z(t_i) =) z_i = 0$ on a discrete time grid $t_i \in \mathbb{T}$.

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference $\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

1. Prior: q-times integrated Wiener process prior:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), Q(h)\right)$$

2. Likelihood: $Z(t) \mid X(t) \sim \delta\left(X^{(1)}(t) - f(X^{(0)}(t), t)\right)$
3. Data: $\mathcal{D}_{\text{PN}} = \{z_i\}$, with $(Z(t_i) =) z_i = 0$ on a discrete time grid $t_i \in \mathbb{T}$.

This describes a state-space model

Probabilistic numerical ODE solutions

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

$$p\left(x(t) \mid x(0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=1}^{N}\right)$$

We want *fast* (approximate) inference $\Rightarrow$ Gaussian filtering and smoothing (it's $\mathcal{O}(N)$!)

1. Prior: q-times integrated Wiener process prior:

$$X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), Q(h)\right)$$

2. Likelihood: $Z(t) \mid X(t) \sim \delta\left(X^{(1)}(t) - f(X^{(0)}(t), t)\right)$
3. Data: $\mathcal{D}_{\text{PN}} = \{z_i\}$, with $(Z(t_i) =)z_i = 0$ on a discrete time grid $t_i \in \mathbb{T}$.

This describes a state-space model $\Rightarrow$ solve with EKF/EKS!

The extended Kalman ODE filter – the state-space model

Bringing the last slides all together

For a given initial value problem $\dot{x}(t) = f(x(t), t)$ on $[0, T]$ with $x(0) = x_0$, we have:

The extended Kalman ODE filter – the state-space model

Bringing the last slides all together

For a given initial value problem $\dot{x}(t) = f(x(t), t)$ on $[0, T]$ with $x(0) = x_0$, we have:

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$
Likelihood / information model:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right)$
Data:	$z_i \triangleq 0, \qquad i = 1, \dots, N.$

The extended Kalman ODE filter – the state-space model

Bringing the last slides all together

For a given initial value problem $\dot{x}(t) = f(x(t), t)$ on $[0, T]$ with $x(0) = x_0$, we have:

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$
Likelihood / information model:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right)$
Data:	$z_i \triangleq 0, \qquad i = 1, \dots, N.$

One thing is still missing:

The extended Kalman ODE filter – the state-space model

Bringing the last slides all together

For a given initial value problem $\dot{x}(t) = f(x(t), t)$ on $[0, T]$ with $x(0) = x_0$, we have:

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$
Likelihood / information model:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right)$
Data:	$z_i \triangleq 0, \qquad i = 1, \ldots, N.$

One thing is still missing:
What about the initial value??

The extended Kalman ODE filter – the state-space model

Bringing the last slides all together

For a given initial value problem $\dot{x}(t) = f(x(t), t)$ on $[0, T]$ with $x(0) = x_0$, we have:

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$
Likelihood / information model:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right)$
Data:	$z_i \triangleq 0, \qquad i = 1, \dots, N.$

One thing is still missing:
What about the initial value??
Just add another "measurement" at $t = 0$:

$$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right), \quad z^{\text{init}} \triangleq x_0.$$

The extended Kalman ODE filter – building blocks

The extended Kalman filter needs these common subroutines

Algorithm 1 Kalman filter prediction

procedure KF_PREDICT(μ, Σ, A, Q)
  $\mu^P \leftarrow A\mu$ // Predict mean
  $\Sigma^P \leftarrow A\Sigma A^\top + Q$ // Predict covariance
  return μ^P, Σ^P
end procedure

Algorithm 2 Extended Kalman filter update

procedure EKF_UPDATE(μ, Σ, h, R, y)
  $\hat{y} \leftarrow h(\mu)$ // evaluate the observation model
  $H \leftarrow J_h(\mu)$ // Jacobian of the observation model
  $S \leftarrow H\Sigma H^\top + R$ // Measurement covariance
  $K \leftarrow \Sigma H^\top S^{-1}$ // Kalman gain
  $\mu^F \leftarrow \mu + K(y - \hat{y})$ // update mean
  $\Sigma^F \leftarrow \Sigma - KSK^\top$ // update covariance
  return μ^F, Σ^F
end procedure

(KF_UPDATE analog but with affine h)

The extended Kalman ODE filter

We can solve ODEs with basically just an extended Kalman filter

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Algorithm 3 The extended Kalman ODE filter

procedure EXTENDED KALMAN ODE FILTER$((\mu_0^-, \Sigma_0^-), (A, Q), (f, x_0), \{t_i\}_{i=1}^N)$

  $\mu_0, \Sigma_0 \leftarrow$ KF_UPDATE$(\mu_0^-, \Sigma_0^-, E_0, \mathbf{0}_{d\times d}, x_0)$ // Initial update to fit the initial value

  for $k \in \{1, \dots, N\}$ **do**

    $h_k \leftarrow t_k - t_{k-1}$ // Step size

    $\mu_k^-, \Sigma_k^- \leftarrow$ 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

    $\mu_k, \Sigma_k \leftarrow$ EKF_UPDATE$(\mu_k^-, \Sigma_k^-, m_k, \mathbf{0}_{d\times d}, \mathbf{0}_d)$ // Extended Kalman filter update

  end for

  return $(\mu_k, \Sigma_k)_{k=1}^N$

end procedure

Recall: The *state* $X(t)$ is a stack of q derivatives $X = \left[X^{(0)}, X^{(1)}, \dots, X^{(q)}\right]^T$.
The projection matrices E_i map X to the i-th derivative: $E_i X = X^{(i)}$.

The extended Kalman ODE filter

We can solve ODEs with basically just an extended Kalman filter

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Algorithm 3 The extended Kalman ODE filter

procedure EXTENDED KALMAN ODE FILTER$((\mu_0^-, \Sigma_0^-), (A, Q), (f, x_0), \{t_i\}_{i=1}^N)$

  $\mu_0, \Sigma_0 \leftarrow$ KF_UPDATE$(\mu_0^-, \Sigma_0^-, E_0, \mathbf{0}_{d\times d}, x_0)$ // Initial update to fit the initial value

  for $k \in \{1, \dots, N\}$ **do**

    $h_k \leftarrow t_k - t_{k-1}$ // Step size

    $\mu_k^-, \Sigma_k^- \leftarrow$ 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

    $\mu_k, \Sigma_k \leftarrow$ EKF_UPDATE$(\mu_k^-, \Sigma_k^-, m_k, \mathbf{0}_{d\times d}, \mathbf{0}_d)$ // Extended Kalman filter update

  end for

  return $(\mu_k, \Sigma_k)_{k=1}^N$

end procedure

Recall: The *state* $X(t)$ is a stack of q derivatives $X = \left[X^{(0)}, X^{(1)}, \dots, X^{(q)}\right]^T$.

The projection matrices E_i map X to the i-th derivative: $E_i X = X^{(i)}$.

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

DEMO TIME: The extended Kalman ODE filter in code

`demo.jl`

Uncertainty calibration or "how to choose prior hyperparameters"

Hyperparameters of the prior have a strong influence on posteriors – so we need to estimate them

- Recall the IWP(q) prior model: $X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), \sigma^2 Q(h)\right)$.
  $\Rightarrow$ The hyperparameter σ directly influences covariances! But what value should it have?

Uncertainty calibration or “how to choose prior hyperparameters”

Hyperparameters of the prior have a strong influence on posteriors – so we need to estimate them

- Recall the IWP(q) prior model: $X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), \sigma^2 Q(h)\right)$.
  $\Rightarrow$ The hyperparameter σ directly influences covariances! But what value should it have?
- **Standard approach**: Maximize the marginal likelihood:

$$\hat{\sigma} = \arg\max p(\mathcal{D}_{\text{PN}} \mid \sigma) = p(z_{1:N} \mid \sigma) = p(z_1 \mid \sigma) \prod_{k=2}^{N} p(z_k | z_{1:k-1}, \sigma).$$

Uncertainty calibration or “how to choose prior hyperparameters”

Hyperparameters of the prior have a strong influence on posteriors – so we need to estimate them

- Recall the IWP(q) prior model: $X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), \sigma^2 Q(h)\right)$.
  $\Rightarrow$ The hyperparameter σ directly influences covariances! But what value should it have?
- **Standard approach**: Maximize the marginal likelihood:

$$\hat{\sigma} = \arg\max p(\mathcal{D}_{\text{PN}} \mid \sigma) = p(z_{1:N} \mid \sigma) = p(z_1 \mid \sigma) \prod_{k=2}^{N} p(z_k | z_{1:k-1}, \sigma).$$

- The EKF provides Gaussian estimates $p(z_k \mid z_{1:k-1}) \approx \mathcal{N}(z_k; \hat{z}_k, S_k)$. This gives a *quasi*-MLE:

$$\hat{\sigma} = \arg\max p(\mathcal{D}_{\text{PN}} \mid \sigma) \approx \arg\max \sum_{k=1}^{N} \log \mathcal{N}\left(z_k; \hat{z}_k, S_k\right).$$

Uncertainty calibration or "how to choose prior hyperparameters"

Hyperparameters of the prior have a strong influence on posteriors – so we need to estimate them

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Recall the IWP(q) prior model: $X(t+h) \mid X(t) \sim \mathcal{N}\left(A(h)X(t), \sigma^2 Q(h)\right)$.
  $\Rightarrow$ The hyperparameter σ directly influences covariances! But what value should it have?
- **Standard approach**: Maximize the marginal likelihood:

$$\hat{\sigma} = \arg\max p(\mathcal{D}_{\text{PN}} \mid \sigma) = p(z_{1:N} \mid \sigma) = p(z_1 \mid \sigma) \prod_{k=2}^{N} p(z_k | z_{1:k-1}, \sigma).$$

- The EKF provides Gaussian estimates $p(z_k \mid z_{1:k-1}) \approx \mathcal{N}\left(z_k; \hat{z}_k, S_k\right)$. This gives a *quasi*-MLE:

$$\hat{\sigma} = \arg\max p(\mathcal{D}_{\text{PN}} \mid \sigma) \approx \arg\max \sum_{k=1}^{N} \log \mathcal{N}\left(z_k; \hat{z}_k, S_k\right).$$

- **In our specific context this can be solved in closed form:**

$$\hat{\sigma}^2 = \frac{1}{Nd} \sum_{i=1}^{N} \left(z_i - \hat{z}_i\right)^\top S_i^{-1} \left(z_i - \hat{z}_i\right).$$

  We don't even need to run the filter again! Just adjust the covariances: (proof: homework)

$$\Sigma_i \leftarrow \hat{\sigma}^2 \cdot \Sigma_i, \qquad \forall i \in \{1, \dots, N\}.$$

DEMO TIME: Calibrated vs uncalibrated posteriors

`demo.jl`

Why?

Why be probabilistic about ODE solutions?

Why be probabilistic about ODE solutions?

- **Uncertainty quantification**:
  The methods provide estimates of their numerical error

Why be probabilistic about ODE solutions?

- **Uncertainty quantification**:
  The methods provide estimates of their numerical error
- **Flexibility / convenience / efficiency**:
  The probabilistic state-space formulation makes it very easy to perform joint inference on various kinds of information
  (this is what we will do next!)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Example 1: **Extending ODE filters to other related problems by adjusting the *information model***

Numerical problem setting: Initial value problem with ODE

$$\dot{x}(t) = f(x(t), t), \quad x(0) = x_0.$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$

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

$$\ddot{x}(t) = f(\dot{x}(t), x(t), t), \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0.$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$

Extending ODE filters to other related differential equation problems

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

Paper: *Bosch, Tronarp, Hennig, AISTATS 2022*

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

$$\ddot{x}(t) = f(\dot{x}(t), x(t), t), \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0.$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(2)}(t_i) - f(X^{(1)}(t_i), X^{(0)}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$
Initial derivative likelihood:	$Z_1^{\text{init}} \mid X(0) \sim \delta\left(X^{(1)}(0)\right),$	$z_1^{\text{init}} \triangleq \dot{x}_0$

Extending ODE filters to other related differential equation problems

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

Paper: *Bosch, Tronarp, Hennig, AISTATS 2022*

Numerical problem setting: Initial value problem with *differential-algebraic equation* (DAE) in mass-matrix form

$$M\dot{x}(t) = f(x(t), t), \quad x(0) = x_0. \qquad \text{(with singular } M\text{)}$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$
ODE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right), \qquad z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right), \qquad z^{\text{init}} \triangleq x_0$

Numerical problem setting: Initial value problem with *differential-algebraic equation* (DAE) in mass-matrix form

$$M\dot{x}(t) = f(x(t), t), \quad x(0) = x_0. \qquad \text{(with singular } M\text{)}$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$
DAE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(MX^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right), \qquad z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right), \qquad z^{\text{init}} \triangleq x_0$

Extending ODE filters to other related differential equation problems

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

Paper: *Bosch, Tronarp, Hennig, AISTATS 2022*

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

$$\dot{x}(t) = f(x(t), t), \quad x(0) = x_0, \quad g(x(t), \dot{x}(t)) = 0.$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$

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

$$\dot{x}(t) = f(x(t), t), \quad x(0) = x_0, \quad g(x(t), \dot{x}(t)) = 0.$$

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

Initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
Prior / dynamics model:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), t_i)\right),$	$z_i \triangleq 0$
Conservation law likelihood:	$Z_i^c(t_i) \mid X(t_i) \sim \delta\left(g(X^{(0)}(t_i), X^{(1)}(t_i))\right),$	$z_i^c \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

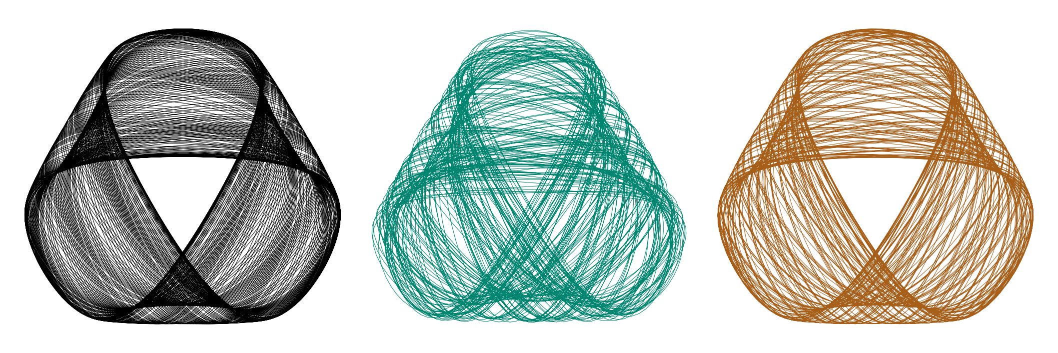
DEMO TIME: Solving a second-order ODE

`demo.jl`

DEMO TIME: Conserved quantities

`henonheiles.mp4`

DEMO TIME: Conserved quantities

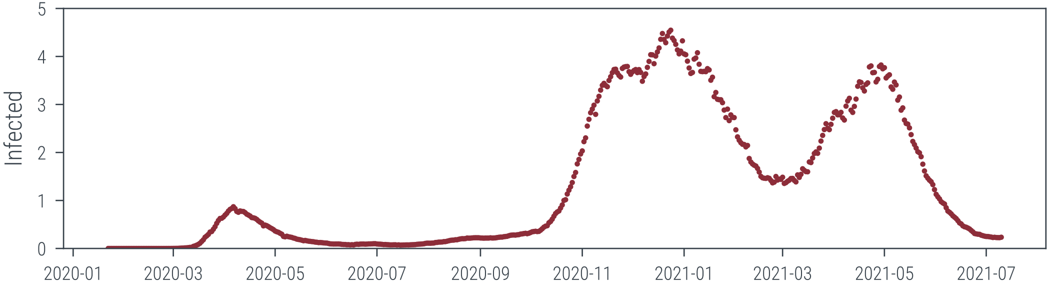


Example 2: **Combine ODEs and GP regression via *latent force inference***

Latent force inference: GP regression on both ODEs and data

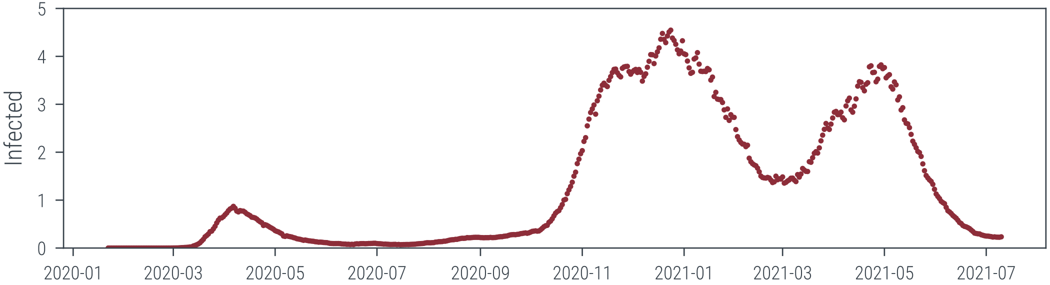
An example we know all too well: COVID-19

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*



ODE dynamics:

$$\frac{d}{dt}x(t) = f(x(t), t)$$



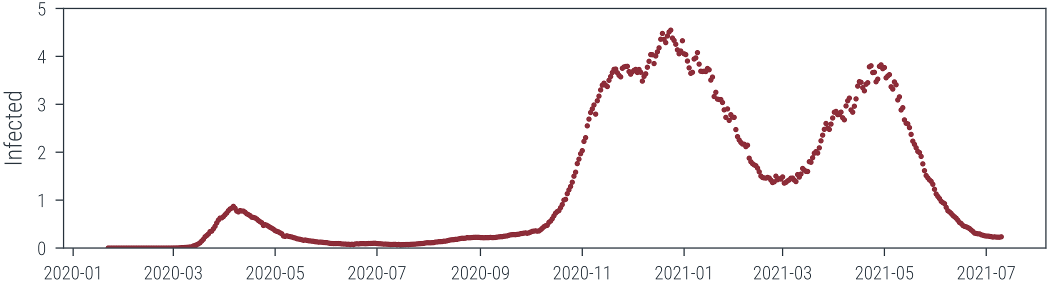
ODE dynamics:

$$\frac{d}{dt}\begin{bmatrix} S(t) \\ I(t) \\ R(t) \\ D(t) \end{bmatrix} = \begin{bmatrix} -\beta \cdot S(t)I(t)/P \\ \beta \cdot S(t)I(t)/P - \gamma I(t) - \eta I(t) \\ \gamma I(t) \\ \eta I(t) \end{bmatrix}$$

Latent force inference: GP regression on both ODEs and data

An example we know all too well: COVID-19

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*



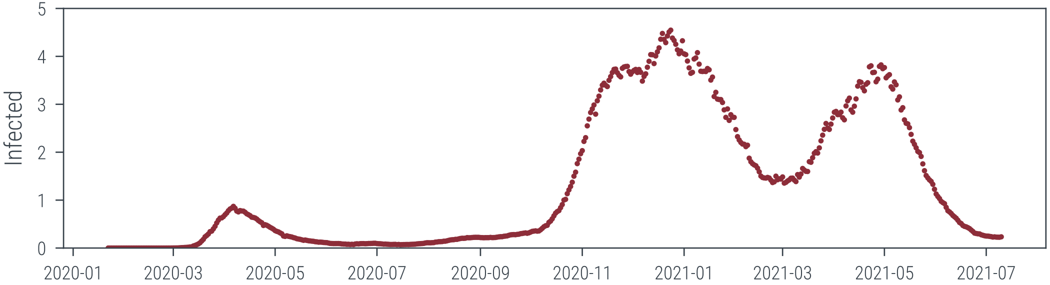
ODE dynamics with time-varying contact rate:

$$\frac{d}{dt}\begin{bmatrix} S(t) \\ I(t) \\ R(t) \\ D(t) \end{bmatrix} = \begin{bmatrix} -\beta(t) \cdot S(t)I(t)/P \\ \beta(t) \cdot S(t)I(t)/P - \gamma I(t) - \eta I(t) \\ \gamma I(t) \\ \eta I(t) \end{bmatrix}$$

Latent force inference: GP regression on both ODEs and data

An example we know all too well: COVID-19

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*



ODE dynamics with time-varying contact rate:

$$\frac{d}{dt}\begin{bmatrix} S(t) \\ I(t) \\ R(t) \\ D(t) \end{bmatrix} = \begin{bmatrix} -\beta(t) \cdot S(t)I(t)/P \\ \beta(t) \cdot S(t)I(t)/P - \gamma I(t) - \eta I(t) \\ \gamma I(t) \\ \eta I(t) \end{bmatrix}$$

Latent force model: Gauss–Markov process

$$\beta(t+h) \mid \beta(t) \sim \mathcal{N}\left(A_\beta(h)\beta(t), Q_\beta(h)\right)$$

Data:

$$y_i \mid x(t_i) \sim \mathcal{N}\left(Hx(t_i), \sigma^2 I\right)$$

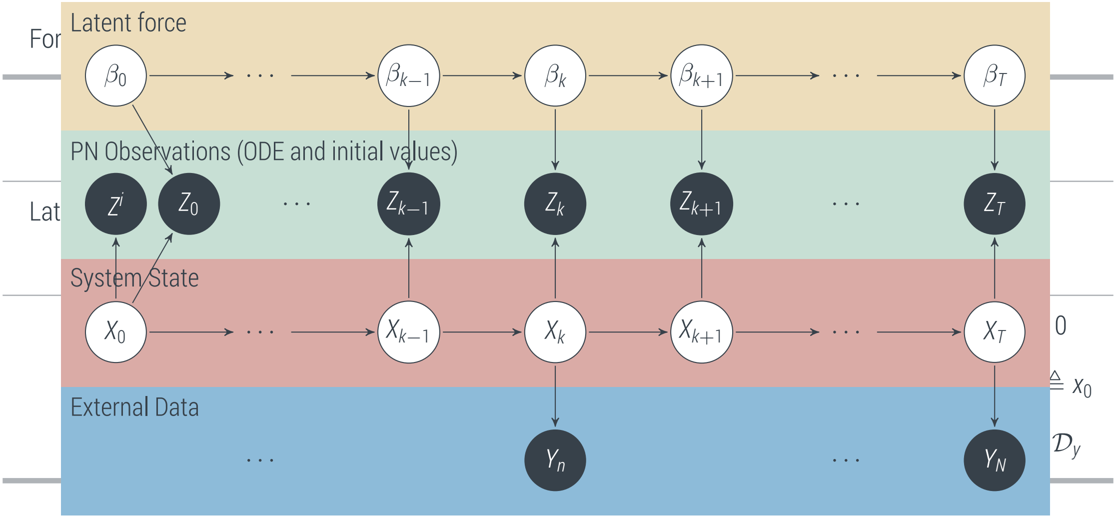
Formally we obtain the **probabilistic state estimation problem:**

State initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
State dynamics:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
Latent force initial distribution:	$\beta(0) \sim \mathcal{N}\left(\mu_0^\beta, \Sigma_0^\beta\right)$	
Latent force dynamics:	$\beta(t+h) \mid \beta(t) \sim \mathcal{N}(A_\beta(h)\beta(t), Q_\beta(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i), \beta(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), \beta(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$
Data likelihood:	$Y_i \mid X(t_i) \sim \mathcal{N}\left(HX^{(0)}(t_i), \sigma^2 I\right),$	$y_i \in \mathcal{D}_y$

Latent force inference: Writing down the state estimation problem

Once again it's just a state estimation problem

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*



Latent force inference: Writing down the state estimation problem

Once again it's just a state estimation problem

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*

Formally we obtain the **probabilistic state estimation problem:**

State initial distribution:	$X(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$	
State dynamics:	$X(t+h) \mid X(t) \sim \mathcal{N}(A(h)X(t), Q(h))$	
Latent force initial distribution:	$\beta(0) \sim \mathcal{N}\left(\mu_0^\beta, \Sigma_0^\beta\right)$	
Latent force dynamics:	$\beta(t+h) \mid \beta(t) \sim \mathcal{N}(A_\beta(h)\beta(t), Q_\beta(h))$	
ODE likelihood:	$Z(t_i) \mid X(t_i), \beta(t_i) \sim \delta\left(X^{(1)}(t_i) - f(X^{(0)}(t_i), \beta(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid X(0) \sim \delta\left(X^{(0)}(0)\right),$	$z^{\text{init}} \triangleq x_0$
Data likelihood:	$Y_i \mid X(t_i) \sim \mathcal{N}\left(HX^{(0)}(t_i), \sigma^2 I\right),$	$y_i \in \mathcal{D}_y$

Latent force inference: Writing down the state estimation problem

Once again it's just a state estimation problem

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*

Simplify by stacking $\tilde{X} = [X, \beta]$:

Initial distribution:	$\tilde{X}(0) \sim \mathcal{N}\left(\tilde{\mu}_0, \tilde{\Sigma}_0\right)$	
Prior / dynamics model:	$\tilde{X}(t+h) \mid \tilde{X}(t) \sim \mathcal{N}\left(\tilde{A}(h)\tilde{X}(t), \tilde{Q}(h)\right)$	
ODE likelihood:	$Z(t_i) \mid \tilde{X}(t_i) \sim \delta\left(E_1\tilde{X}(t_i) - f(E_0\tilde{X}(t_i), E_\beta\tilde{X}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid \tilde{X}(0) \sim \delta\left(E_0\tilde{X}(0)\right),$	$z^{\text{init}} \triangleq x_0$
Data likelihood:	$Y_i \mid \tilde{X}(t_i) \sim \mathcal{N}\left(HE_0\tilde{X}(t_i), \sigma^2 I\right),$	$y_i \in \mathcal{D}_y$

with $E_0\tilde{X} := X^{(0)}$, $E_1\tilde{X} := X^{(1)}$, $E_\beta\tilde{X} := \beta$.

Latent force inference: Writing down the state estimation problem

Once again it's just a state estimation problem

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*

Simplify by stacking $\tilde{X} = [X, \beta]$:

Initial distribution:	$\tilde{X}(0) \sim \mathcal{N}\left(\tilde{\mu}_0, \tilde{\Sigma}_0\right)$	
Prior / dynamics model:	$\tilde{X}(t+h) \mid \tilde{X}(t) \sim \mathcal{N}\left(\tilde{A}(h)\tilde{X}(t), \tilde{Q}(h)\right)$	
ODE likelihood:	$Z(t_i) \mid \tilde{X}(t_i) \sim \delta\left(E_1\tilde{X}(t_i) - f(E_0\tilde{X}(t_i), E_\beta\tilde{X}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid \tilde{X}(0) \sim \delta\left(E_0\tilde{X}(0)\right),$	$z^{\text{init}} \triangleq x_0$
Data likelihood:	$Y_i \mid \tilde{X}(t_i) \sim \mathcal{N}\left(HE_0\tilde{X}(t_i), \sigma^2 I\right),$	$y_i \in \mathcal{D}_y$

with $E_0\tilde{X} := X^{(0)}$, $E_1\tilde{X} := X^{(1)}$, $E_\beta\tilde{X} := \beta$.

Again: **This is just a state-space model**

Latent force inference: Writing down the state estimation problem

Once again it's just a state estimation problem

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*

Simplify by stacking $\tilde{X} = [X, \beta]$:

Initial distribution:	$\tilde{X}(0) \sim \mathcal{N}\left(\tilde{\mu}_0, \tilde{\Sigma}_0\right)$	
Prior / dynamics model:	$\tilde{X}(t+h) \mid \tilde{X}(t) \sim \mathcal{N}\left(\tilde{A}(h)\tilde{X}(t), \tilde{Q}(h)\right)$	
ODE likelihood:	$Z(t_i) \mid \tilde{X}(t_i) \sim \delta\left(E_1\tilde{X}(t_i) - f(E_0\tilde{X}(t_i), E_\beta\tilde{X}(t_i), t_i)\right),$	$z_i \triangleq 0$
Initial value likelihood:	$Z^{\text{init}} \mid \tilde{X}(0) \sim \delta\left(E_0\tilde{X}(0)\right),$	$z^{\text{init}} \triangleq x_0$
Data likelihood:	$Y_i \mid \tilde{X}(t_i) \sim \mathcal{N}\left(HE_0\tilde{X}(t_i), \sigma^2 I\right),$	$y_i \in \mathcal{D}_y$

with $E_0\tilde{X} := X^{(0)}$, $E_1\tilde{X} := X^{(1)}$, $E_\beta\tilde{X} := \beta$.

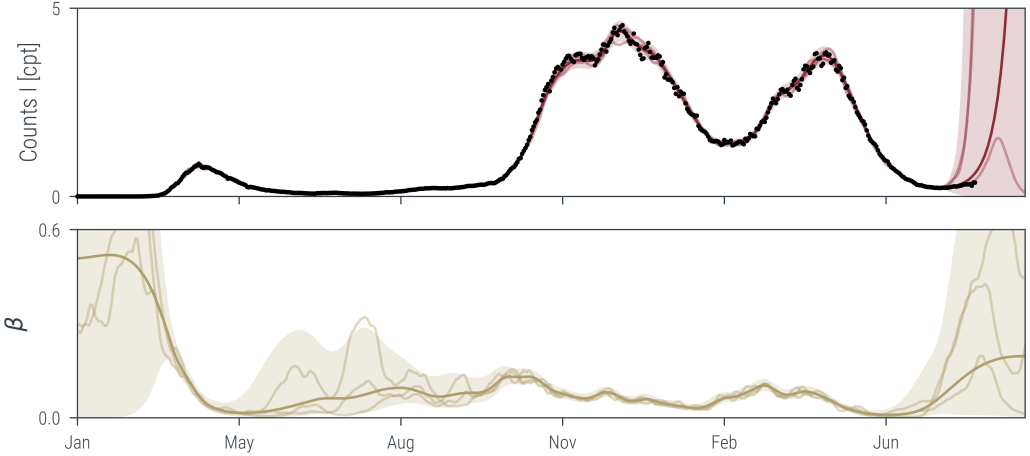
Again: **This is just a state-space model $\Rightarrow$ inference with EKF/EKS!**

Latent force inference: Results

Posteriors over infections and contact rates *in a single forward-backward pass*

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*

The extended Kalman smoother returns probabilistic estimates for all states (S, I, R, D, β):

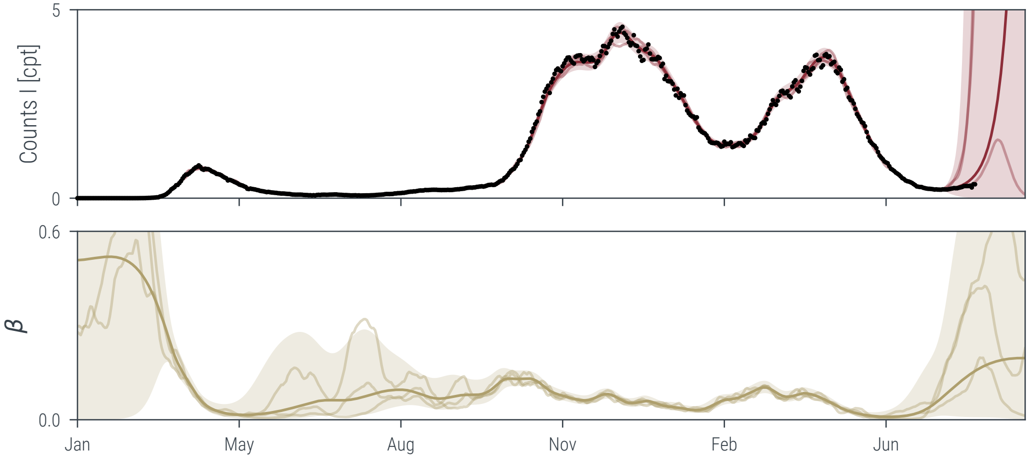


Latent force inference: Results

Posteriors over infections and contact rates *in a single forward-backward pass*

Paper: *Schmidt, Krämer, Hennig, NeurIPS 2021*

The extended Kalman smoother returns probabilistic estimates for all states (S, I, R, D, β):



$\Rightarrow$ Probabilistic estimates of a latent force *in a single forward-backward pass*!

Summary

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- *ODE solving is state estimation*
  $\Rightarrow$ treat ODEs as state estimation problems!
- **We can solve ODEs with Bayesian filtering and smoothing**
  $\Rightarrow$ *"ODE filters"*
- *Flexible information operators*:
  Easily adjust the model to solve other numerical problems, *with essentially the same algorithm*!
- *Latent force inference*:
  Filters enable *efficient* joint inference on both ODEs and data.

Please cite this course, as

```
@techreport{NoML22,
    title = {Numerics of Machine Learning},
    author = {N. Bosch and J. Grosse
    and P. Hennig and A. Kristiadi
    and M. Pförtner and J. Schmidt
    and F. Schneider and L. Tatzel
    and J. Wenger},
    series = {Lecture Notes in Machine Learning},
    year = {2022},
    institution = {Tübingen AI Center},
}
```

Bayesian filtering and smoothing is the right framework for modeling dynamical systems in a modular and data-centric fashion.

Next week: ***Partial* Differential Equations**!