Robust Parameter Inference in ODEs via Physics-Enhanced Gaussian Process Regression

ProbNum 24

Nathanael Bosch

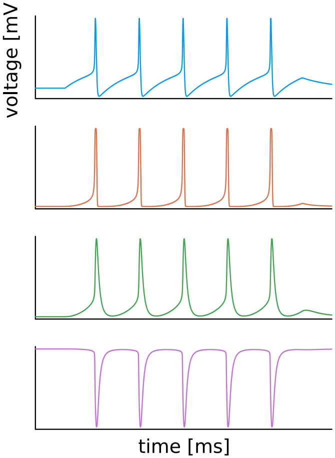
16. July 2024

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

The ODE parameter inference problem

- Initial value problem:

$$\dot{y}(t) = f_\theta(y(t), t), \quad t \in [0, T], \qquad y(0) = y_{0,\theta}$$



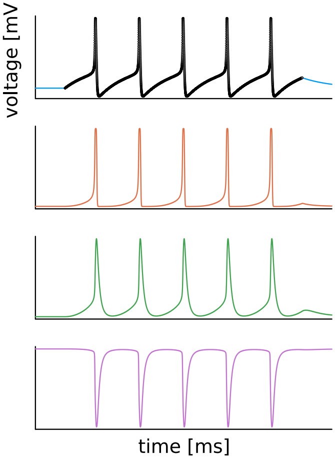
The ODE parameter inference problem

- Initial value problem:

$$\dot{y}(t) = f_\theta(y(t), t), \quad t \in [0, T], \qquad y(0) = y_{0,\theta}$$

- Observations:

$$u_i = Hy(t_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, R_\theta)$$



The ODE parameter inference problem

- Initial value problem:

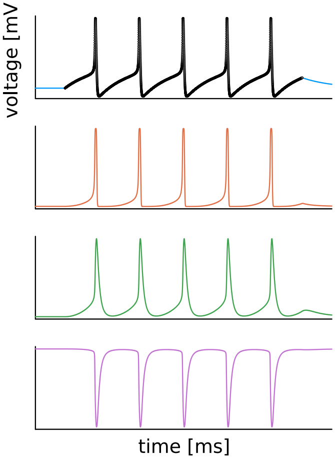
$$\dot{y}(t) = f_\theta(y(t), t), \quad t \in [0, T], \qquad y(0) = y_{0,\theta}$$

- Observations:

$$u_i = Hy(t_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, R_\theta)$$

- Goal:

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta) p(\theta)$$



The ODE parameter inference problem

- Initial value problem:

$$\dot{y}(t) = f_\theta(y(t), t), \quad t \in [0, T], \qquad y(0) = y_{0,\theta}$$

- Observations:

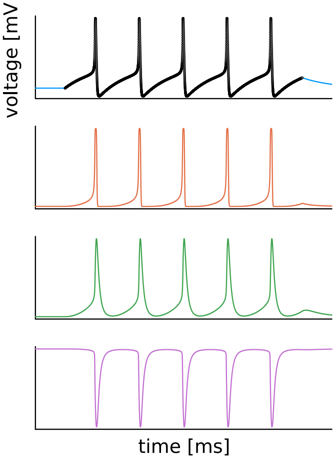
$$u_i = Hy(t_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, R_\theta)$$

- Goal:

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta) p(\theta)$$

- Maximum likelihood, maximum-a-posteriori and MCMC require the marginal likelihood:

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta)$$



Investigating the marginal likelihood

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))} \underbrace{p(y(t_{1:N}) \mid \theta)} \mathrm{d}y(t_{1:N})$$

Investigating the marginal likelihood

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)} \mathrm{d}y(t_{1:N})$$

Investigating the marginal likelihood

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)}_{???} \mathrm{d}y(t_{1:N})$$

Investigating the marginal likelihood

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)}_{???} \mathrm{d}y(t_{1:N})$$

- "$y(t)$ given θ" is fully specified via the ODE
  $\Rightarrow p(y(t_{1:N}) \mid \theta) = \delta(y(t_{1:N}) - y^*(t_{1:N}))$

Investigating the marginal likelihood

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)}_{???} \mathrm{d}y(t_{1:N})$$

- "$y(t)$ given θ" is fully specified via the ODE
  $\Rightarrow p(y(t_{1:N}) \mid \theta) = \delta(y(t_{1:N}) - y^*(t_{1:N}))$
- But the true ODE solution $y^*(t_{1:N})$ is intractable!

Investigating the marginal likelihood

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)}_{???} \mathrm{d}y(t_{1:N})$$

- "$y(t)$ given θ" is fully specified via the ODE
  $\Rightarrow p(y(t_{1:N}) \mid \theta) = \delta(y(t_{1:N}) - y^*(t_{1:N}))$
- But the true ODE solution $y^*(t_{1:N})$ is intractable!
- Let's approximate:

$$\delta(y(t_{1:N}) - y^*_\theta(t_{1:N})) \approx$$

Investigating the marginal likelihood

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)}_{???} \, \mathrm{d}y(t_{1:N})$$

- "$y(t)$ given θ" is fully specified via the ODE
  $\Rightarrow p(y(t_{1:N}) \mid \theta) = \delta(y(t_{1:N}) - y^*(t_{1:N}))$
- But the true ODE solution $y^*(t_{1:N})$ is intractable!
- Let's approximate:

$$\delta(y(t_{1:N}) - y^*_\theta(t_{1:N})) \approx \delta(y(t_{1:N}) - \hat{y}_\theta(t_{1:N})) \qquad \text{(the classic numerical approach)}$$

Investigating the marginal likelihood

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \theta)}_{???} \mathrm{d}y(t_{1:N})$$

- "$y(t)$ given θ" is fully specified via the ODE
  $\Rightarrow p(y(t_{1:N}) \mid \theta) = \delta(y(t_{1:N}) - y^*(t_{1:N}))$
- But the true ODE solution $y^*(t_{1:N})$ is intractable!
- Let's approximate:

$$\delta(y(t_{1:N}) - y^*_\theta(t_{1:N})) \approx p_{\text{PN}}(y(t_{1:N}) \mid \theta) \qquad \text{(the probabilistic numerical approach)}$$

Probabilistic numerical ODE solvers

Filtering-based probabilistic numerical ODE solvers

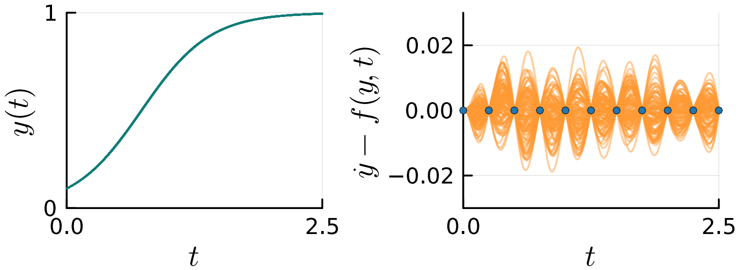
$$p\left(y(t) \,\middle|\, y(0) = y_0, \{\dot{y}(t_n) = f(y(t_n), t_n)\}_{n=1}^{N}\right)$$

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

Filtering-based probabilistic numerical ODE solvers

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

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



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

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:**

Filtering-based probabilistic numerical ODE solvers

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

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

Filtering-based probabilistic numerical ODE solvers

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

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

$$x(t+h) \mid x(t) \sim \mathcal{N}(A(h)x(t), Q(h)),$$
$$y(t) = E_0 x(t), \qquad \dot{y}(t) = E_1 x(t)$$

Filtering-based probabilistic numerical ODE solvers

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

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

$$x(t+h) \mid x(t) \sim \mathcal{N}(A(h)x(t), Q(h)),$$
$$y(t) = E_0 x(t), \qquad \dot{y}(t) = E_1 x(t)$$

- **Likelihood:** (aka "observation model" or "information operator")

Filtering-based probabilistic numerical ODE solvers

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

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

$$x(t+h) \mid x(t) \sim \mathcal{N}(A(h)x(t), Q(h)),$$
$$y(t) = E_0 x(t), \qquad \dot{y}(t) = E_1 x(t)$$

- **Likelihood:** (aka "observation model" or "information operator")

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

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

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

$$x(t+h) \mid x(t) \sim \mathcal{N}\left(A(h)x(t), Q(h)\right),$$
$$y(t) = E_0 x(t), \qquad \dot{y}(t) = E_1 x(t)$$

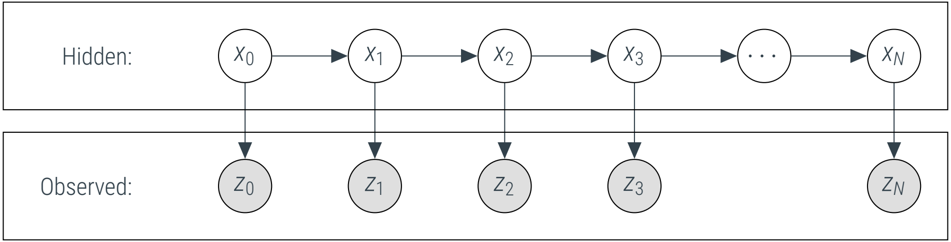
- **Likelihood:** (aka “observation model” or “information operator”)

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

- **Inference:** Bayesian filtering and smoothing
  Extended Kalman filter, unscented Kalman filter, particle filters, ... (+ smoothers)

Probabilistic ODE solvers: the state-estimation problem

This is the actual state estimation problem that we solve



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: $z(t_i) \mid x(t_i) \sim \delta\left(z(t_i); E_1 x(t_i) - f(E_0 x(t_i), t_i)\right), \quad z_i \triangleq 0$

Initial value likelihood: $z^{\text{init}} \mid x(0) \sim \delta\left(z^{\text{init}}; E_0 x(0) - y_0\right), \quad z^{\text{init}} \triangleq 0$

Probabilistic ODE solvers in pseudo code

We can solve ODEs with basically just an extended Kalman filter

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Algorithm The extended Kalman ODE filter

procedure EXTENDED KALMAN ODE FILTER$((\mu_0^-, \Sigma_0^-), (A, Q), (f, y_0), \{t_i\}_{i=1}^N)$

    $\mu_0, \Sigma_0 \leftarrow \text{KF\_UPDATE}(\mu_0^-, \Sigma_0^-, E_0, 0_{d\times d}, y_0)$ // 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

        $\mu_k, \Sigma_k \leftarrow \text{EKF\_UPDATE}(\mu_k^-, \Sigma_k^-, m_k, 0_{d\times d}, \vec{0}_d)$ // Extended Kalman filter update

    end for

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

end procedure

https://github.com/nathanaelbosch/probnumspringschool2024-tutorial

Computing the PN-approximated marginal likelihood

How to compute the PN-approximated margial likelihood

It's just another filtering problem

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \mathcal{D}_{\text{PN}}, \theta)}_{\text{PN posterior}} \mathrm{d}y(t_{1:N})$$

How to compute the PN-approximated margial likelihood

It's just another filtering problem

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \mathcal{D}_{\text{PN}}, \theta)}_{\text{PN posterior}} \mathrm{d}y(t_{1:N})$$

Filtering posteriors have a recursive, linear Gaussian, backward-in-time representation:

$$p(x(t_{1:N}) \mid \mathcal{D}_{\text{PN}}, \theta) = \mathcal{N}\left(x(t_N); \mu_N^F, \Sigma_N^F\right) \prod_{t=1}^{N-1} \mathcal{N}(x(t_n); G_n x(t_{n+1}) + d_n, \Lambda_n);$$

marginalizing this posterior is exactly what a *smoother* does.

How to compute the PN-approximated margial likelihood

It's just another filtering problem

$$\mathcal{M}(\theta) = p(\mathcal{D} \mid \theta) = \int \underbrace{p(\mathcal{D} \mid y(t_{1:N}))}_{\text{Gaussian likelihood}} \underbrace{p(y(t_{1:N}) \mid \mathcal{D}_{\text{PN}}, \theta)}_{\text{PN posterior}} \mathrm{d}y(t_{1:N})$$

Filtering posteriors have a recursive, linear Gaussian, backward-in-time representation:

$$p(x(t_{1:N}) \mid \mathcal{D}_{\text{PN}}, \theta) = \mathcal{N}\left(x(t_N); \mu_N^F, \Sigma_N^F\right) \prod_{t=1}^{N-1} \mathcal{N}(x(t_n); G_n x(t_{n+1}) + d_n, \Lambda_n);$$

marginalizing this posterior is exactly what a *smoother* does.

State-space model:

Initial distribution: $x(t_N) \sim \mathcal{N}\left(x(t_N); \mu_N^F, \Sigma_N^F\right)$

Dynamics model: $x(t_{n-1}) \mid x(t_n) \sim \mathcal{N}(x(t_n); G_n x(t_{n+1}) + d_n, \Lambda_n)$

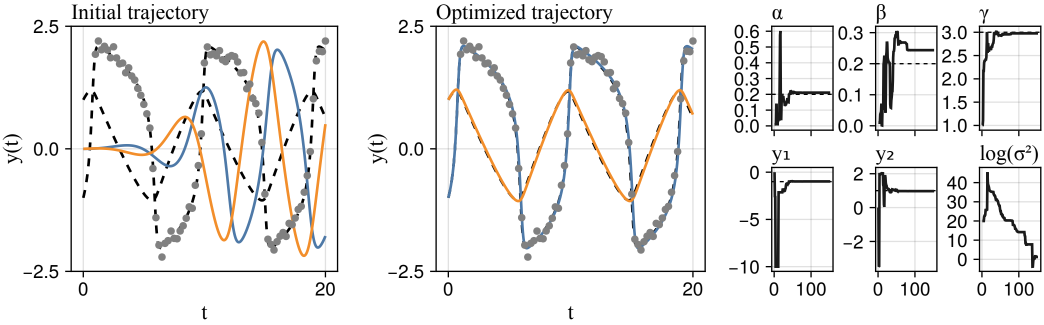
Data likelihood: $u_n \mid x(t_n) \sim \mathcal{N}(x(t_n); H E_0 x(t_n), R_\theta)$

Resulting algorithm:

1. Run filter forwards to compute $p(y(t_{1:N}) \mid \mathcal{D}_{\text{PN}}, \theta)$
2. Run filter backwards to compute the marginal likelihood $\mathcal{M}(\theta)$

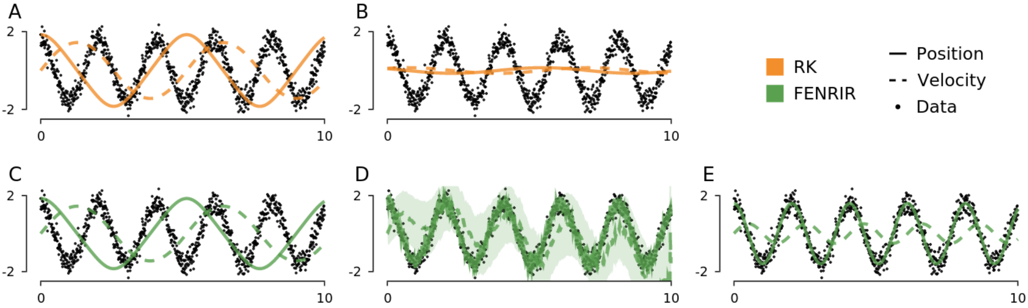
MLE parameter inference with FENRIR

It works!



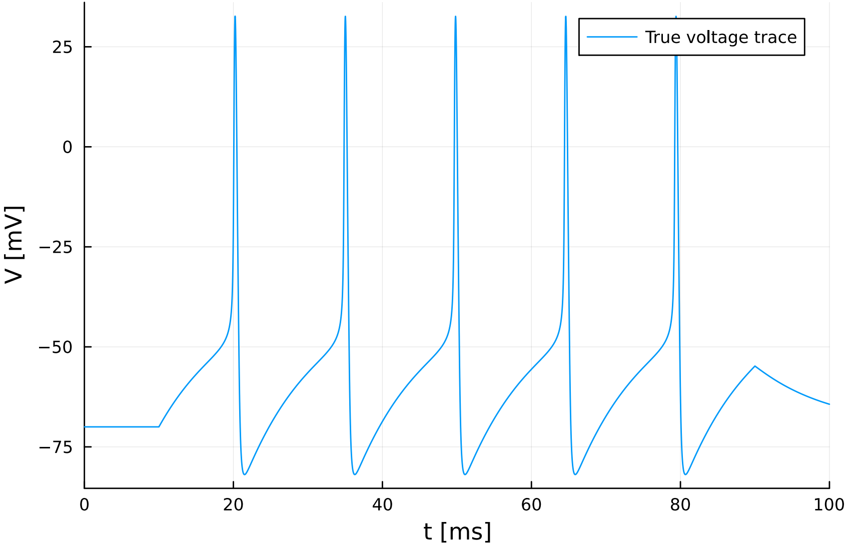
MLE parameter inference with FENRIR

The algorithm is quite robust to local optima

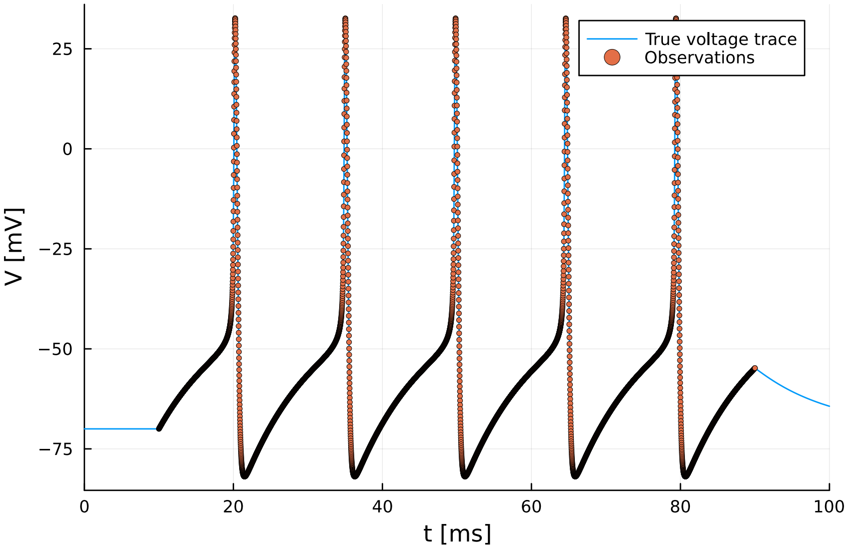


Parameter inference in Hodgkin-Huxley ODEs

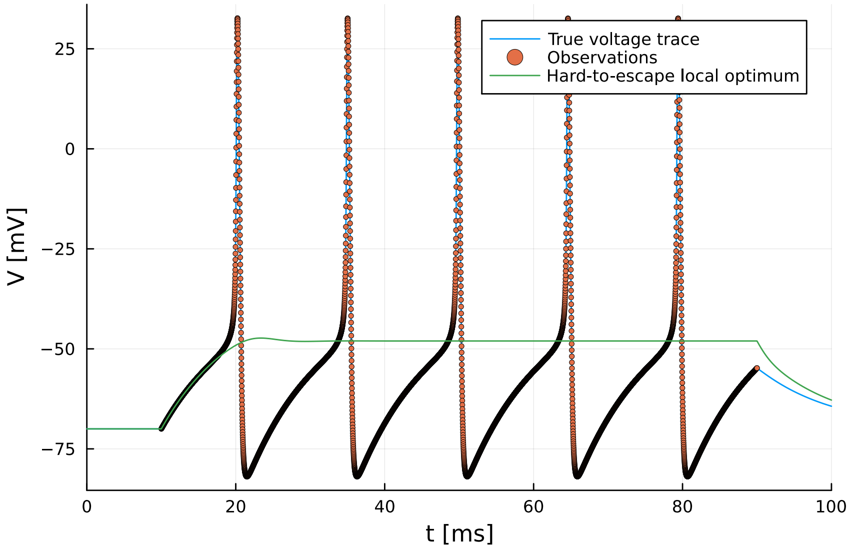
Parameter inference in Hodgkin-Huxley ODEs



Parameter inference in Hodgkin-Huxley ODEs



Parameter inference in Hodgkin-Huxley ODEs



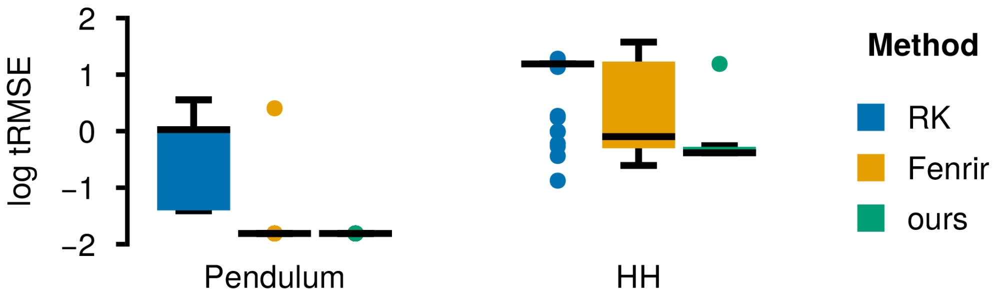
MLE parameter inference with FENRIR on the Hodgkin-Huxley model

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

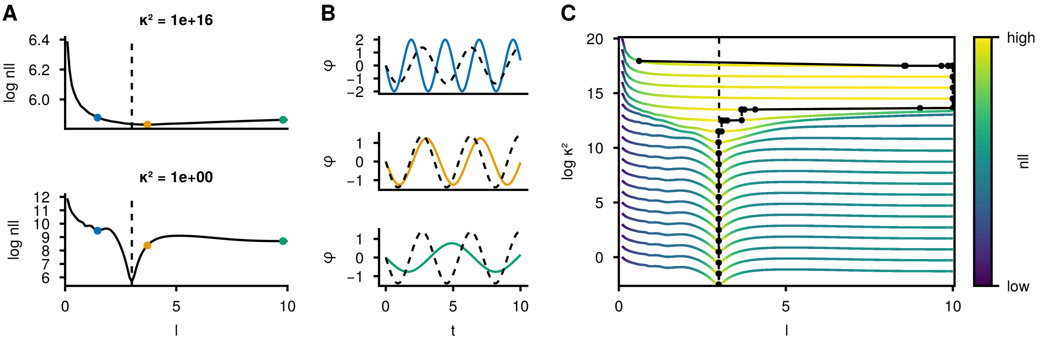
It works, but clearly not as well as for the simple pendulum problem

MLE parameter inference with FENRIR on the Hodgkin-Huxley model

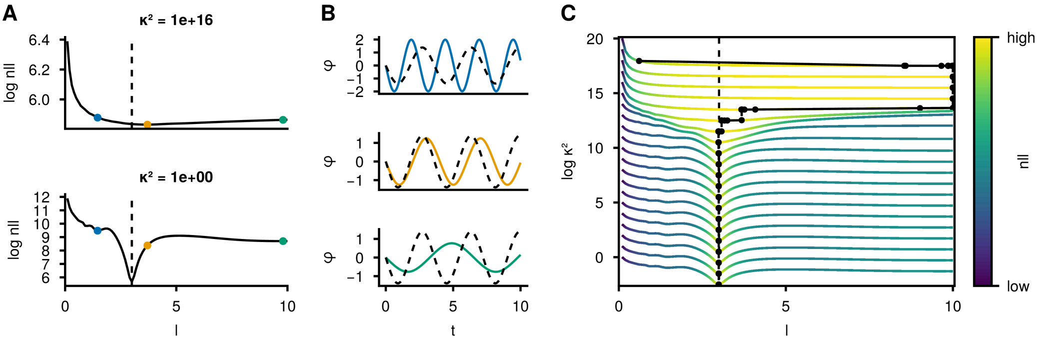
It works, but clearly not as well as for the simple pendulum problem



Idea: Diffusion Tempering



Idea: Diffusion Tempering

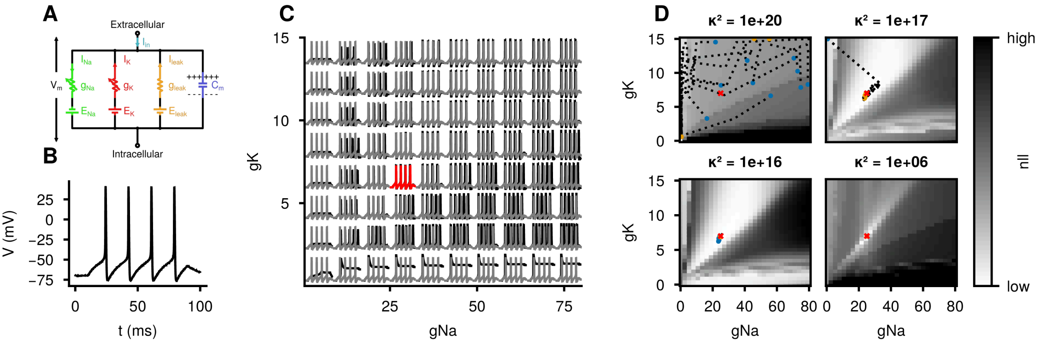


Algorithm: Start with an initial parameter guess θ_0. Then for $i = 1, \ldots, M$ solve a sequence of MLE optimization problems

$$\theta_i = \arg\max \mathcal{M}(\theta, \Gamma(i)) = \text{OPTIMIZE}(\mathcal{M}; \theta_{\text{init}} = \theta_{i-1}, \sigma = \Gamma(i)). \quad (1)$$

FENRIR + diffusion tempering on the Hodgkin-Huxley ODE

It works!



FENRIR + diffusion tempering on the Hodgkin-Huxley ODE

It works!

HH	1	FENRIR	.	.	.	.	0.68	.	.	.	.
HH	1	RK	43.30	43.45	.	.	0.57	.	.	.	.
HH	1	OURS	.	.	0.00	0.00	1.00	0.43	0.02	1.00	0.00
HH	1	OURS+	.	.	0.00	0.00	1.00	.	.	1.00	0.00
HH	2	FENRIR	.	.	.	.	0.75	.	.	.	.
HH	2	RK	54.02	62.60	.	.	0.72	.	.	.	.
HH	2	OURS	.	.	0.00	0.00	1.00	0.42	0.04	2.00	0.00
HH	2	OURS+	.	.	.	.	0.96	.	.	.	.
HH	3	FENRIR	122.15	49.74	.	.	0.51	.	.	.	.
HH	3	RK	.	.	.	.	0.03	.	.	.	.
HH	3	OURS	.	.	0.01	0.10	0.99	0.60	1.51	2.97	0.30
HH	6	FENRIR	108.06	108.49	.	.	0.00	.	.	.	.
HH	6	RK	.	.	.	.	0.00	.	.	.	.
HH	6	OURS	.	.	10.36	7.72	0.00	15.20	5.41	1.21	0.46
HH	4	FENRIR	.	.	.	.	0.68	.	.	.	.
HH	4	RK	136.50	200.20	.	.	0.50	.	.	.	.
HH	4	OURS	.	.	0.00	0.00	1.00	0.60	0.01	4.00	0.00
HH	6	FENRIR	221.28	144.56	.	.	0.50	.	.	.	.
HH	6	RK	.	.	.	.	0.00	.	.	.	.
HH	6	OURS	.	.	0.12	0.32	0.88	3.01	6.70	5.28	1.96

An alternative way to compute the PN-approximated marginal likelihood

An alternative PN likelihood approximation method: DALTON

"Data-Adaptive Probabilistic Likelihood Approximation" [Wu and Lysy, 2024]

$$p(\mathcal{D}_{\text{data}} \mid \theta, \mathcal{D}_{\text{PN}}) = \frac{p(\mathcal{D}_{\text{data}}, \mathcal{D}_{\text{PN}} \mid \theta)}{p(\mathcal{D}_{\text{PN}} \mid \theta)}$$

An alternative PN likelihood approximation method: DALTON

"Data-Adaptive Probabilistic Likelihood Approximation" [Wu and Lysy, 2024]

$$p(\mathcal{D}_{\text{data}} \mid \theta, \mathcal{D}_{\text{PN}}) = \frac{p(\mathcal{D}_{\text{data}}, \mathcal{D}_{\text{PN}} \mid \theta)}{p(\mathcal{D}_{\text{PN}} \mid \theta)}$$

To compute:

- $p(\mathcal{D}_{\text{PN}} \mid \theta)$: Standard EKF with PN likelihood
- $p(\mathcal{D}_{\text{data}}, \mathcal{D}_{\text{PN}} \mid \theta)$: EKF with two likelihood models for "PN observations" and the actual data

An alternative PN likelihood approximation method: DALTON

"Data-Adaptive Probabilistic Likelihood Approximation" [Wu and Lysy, 2024]

$$p(\mathcal{D}_{\text{data}} \mid \theta, \mathcal{D}_{\text{PN}}) = \frac{p(\mathcal{D}_{\text{data}}, \mathcal{D}_{\text{PN}} \mid \theta)}{p(\mathcal{D}_{\text{PN}} \mid \theta)}$$

To compute:

- $p(\mathcal{D}_{\text{PN}} \mid \theta)$: Standard EKF with PN likelihood
- $p(\mathcal{D}_{\text{data}}, \mathcal{D}_{\text{PN}} \mid \theta)$: EKF with two likelihood models for "PN observations" and the actual data

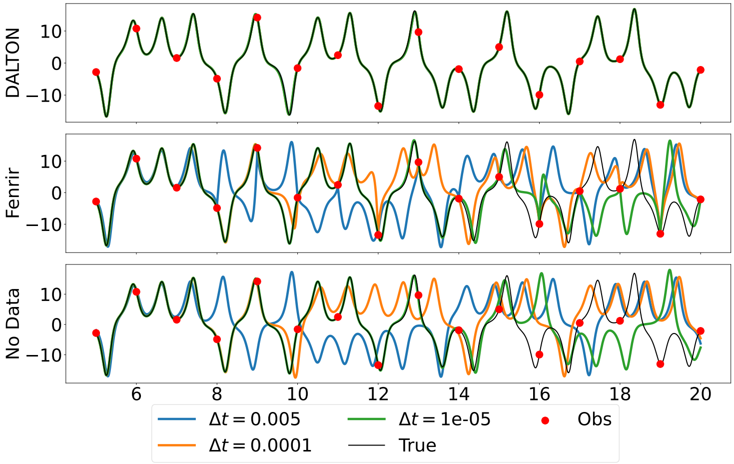
$\Rightarrow$ Run two filters!

FENRIR vs DALTON: Lorenz63

Updating on data in the forward pass can severly improve the ODE solution

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

[Wu and Lysy, 2024]

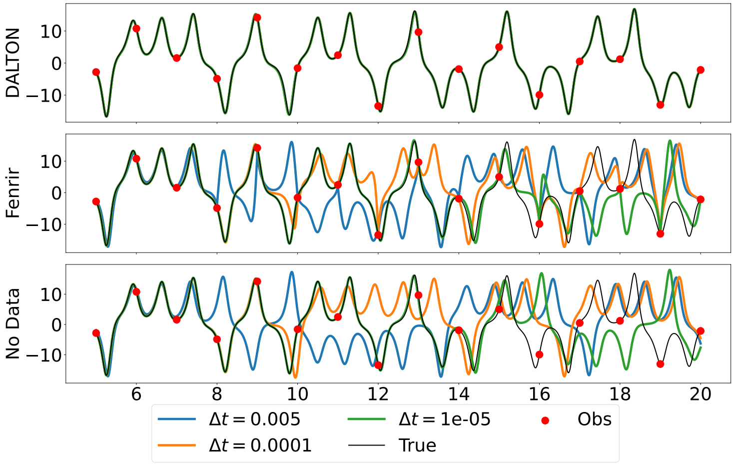


Pros / cons:

FENRIR vs DALTON: Lorenz63

Updating on data in the forward pass can severly improve the ODE solution

[Wu and Lysy, 2024]



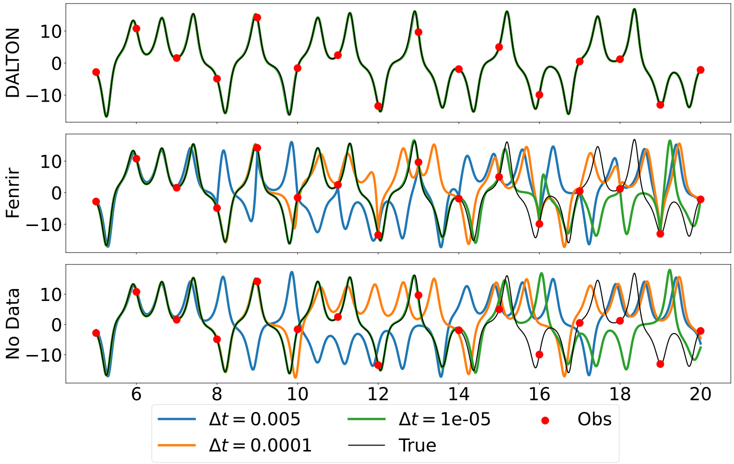
Pros / cons:

- + Better performace for chaotic systems

FENRIR vs DALTON: Lorenz63

Updating on data in the forward pass can severly improve the ODE solution

[Wu and Lysy, 2024]



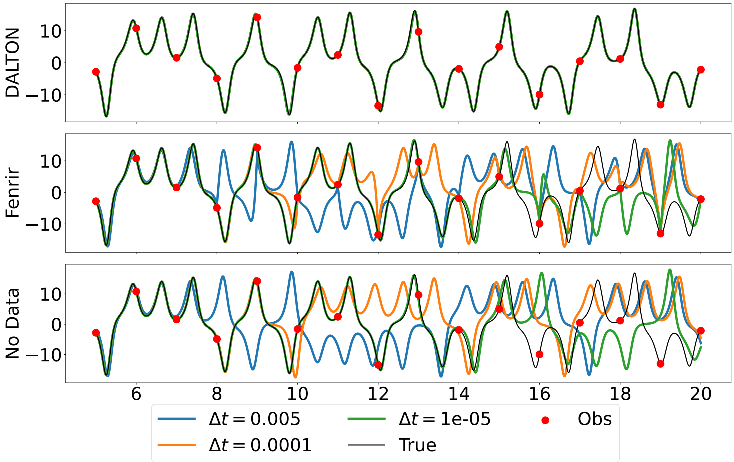
Pros / cons:

- \+ Better performace for chaotic systems
- \- Needs to solve the ODE two times

FENRIR vs DALTON: Lorenz63

Updating on data in the forward pass can severly improve the ODE solution

[Wu and Lysy, 2024]



Pros / cons:

- + Better performace for chaotic systems
- - Needs to solve the ODE two times
- + Computationally cheaper as it does not require smoothing!

Summary

- Parameter inference in ODEs requires computing a marginal likelihood
- Use filtering-based probabilistic numerical ODE solvers to approximate it
- **Being probabilistic can help escape local optima**

Software

`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`

Other topic I'm excited about: *Probabilistic numerics for parallel-in-time ODE solving!*

Bibliography I

- Wu, M. and Lysy, M. (2024).
  Data-adaptive probabilistic likelihood approximation for ordinary differential equations.
  In Dasgupta, S., Mandt, S., and Li, Y., editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1018–1026. PMLR.