Graduate Training Centre of Neuroscience
Neural Information Processing
University of Tübingen

Comparing the effect of different priors on simulations of Hodgkin-Huxley dynamics with probabilistic solvers

Report presented by:
Sofiya Garkot

The rotation was supervised by:
Nathanael Bosch, Prof. Philipp Hennig, Department of Computer Science
Jonas Beck, Prof. Philipp Berens, Hertie Institute for AI in Brain Health

Duration of the lab rotation: 27/11/2023 - 26/02/2024

Abstract

Fast and accurate simulations of Hodgkin-Huxley-type neurons are essential for identifying neuronal properties responsible for the observed spiking activity. They facilitate understanding of connection between ion channel properties and neuronal responses. Traditionally, the solution to Hodgkin-Huxley system has relied on classical numerical integrators. These methods offer deterministic point estimates at each step of the solution process, yet they overlook the inherent uncertainties present in computational approximations. In contrast, recently developed probabilistic solvers provide a probability distribution over the solution, incorporating the uncertainty that arises from computation.

Adapting a probabilistic solver to the specifics of an ODE with priors has shown to improve its stability and efficiency. One way to leverage the knowledge about an ODE is to select a more suitable form of covariance functions in the prior. Alternatively, one can choose a prior that would match the support of modeled dynamical variables using link functions.

The results indicate that optimizing the parametrization of Matérn and Integrated Ornstein-Uhlenbeck priors did not significantly reduce the error compared to the standard choice of the prior - Integrated Wiener Process. Using a prior with a constrained range, on the other hand, resulted in a decrease in error at a specified step size, allowing for larger steps to be taken while maintaining the same error.

Contents

1 Introduction

Biophysical models of neurons are of central importance in computational neuroscience since they allow us to understand the mechanisms by which neurons integrate and propagate information. One of the most fundamental models in neuroscience was developed in 1952 by Alan Hodgkin and Andrew Huxley [1]. The authors used a voltage clamp to discover the mechanisms of action potential generation in the squid giant axon. The proposed mathematical formulation links the microscopic level of ion channels to the macroscopic level of currents and action potentials. The model is adaptable, allowing for parameter adjustments, the inclusion of additional ion channel populations, and the modeling of complex dendritic and axonal geometries through multi-compartment models [2].

The propagation of an action potential is described by a non-linear system of ordinary differential equations (ODEs) that cannot be solved analytically. The existing software packages that simulate Hodgkin-Huxley neurons [3–6] approximate a solution with classical numerical integrators, which provide point estimates of solutions with global error estimates.

Numerical solvers typically balance accuracy with computational time, leading researchers to seek solvers that leverage the structure of ODEs to minimize computation speed at a given error tolerance. For instance, Chen et al.[7] applied splitting and composition methods to the Hodgkin-Huxley ODE, demonstrating the preservation of dynamical system qualities, such as limit cycles, even at larger integration steps. Similarly, Börgers and Nectow [8] used the exponential midpoint method to increase a step size up to 1 ms, maintaining a stable solution.

In contrast to classical solvers, recently developed probabilistic ODE solvers return a distribution of solutions at each simulation step, providing interpretable uncertainty estimates. There are two classes of probabilistic ODE solvers: sampling-based solvers [9, 10] and filtering-based solvers [11–13]. Sampling-based probabilistic solvers have previously been applied to neuronal modeling [14]. The authors showed the benefit of using probabilistic solvers for neuronal spiking models. The returned uncertainty offers the advantage of providing confidence intervals for the quantities of interest, such as spike times.

Filtering-based probabilistic solvers formulate the inference of solution as a regression problem in a Bayesian state-space model. Therefore, they allow parametrization via the choice of prior and likelihood models for the solution. It has been shown that adapting a probabilistic solver to the structure of an ODE using prior can improve stability and efficiency of the solver [15, 16]. However, the effect of choosing a different prior is not yet quantified for the Hodgkin-Huxley model.

Another way to adapt a probabilistic solver to the knowledge of an ODE is to use link functions. They adapt the variance of a prior to the range of modeled dynamical variables. Link functions have previously been explored for other models [17]. However, their effect on the accuracy of the solution has not been quantified.

The main objective of this essay is to investigate whether specific priors are making the computation of a solution to Hodgkin-Huxley ODE more effective than the benchmark. The results showed that the structure of the covariance matrix in the Matérn and Integrated Ornstein-Uhlenbeck priors does not significantly influence the efficacy of the solver compared to the baseline Integrated Wiener process. At the same time, priors that have bounded variance using link functions significantly benefit a probabilistic solver.

2 Methods

2.1 Hodgkin-Huxley model

The non-linear system of ODEs in the Hodgkin-Huxley model has the following form:

$$C_M \frac{dV}{dt} = I_e(t) - \overline{g}_{Na} m^3 h (V - E_{Na}) - \overline{g}_K n^4 (V - E_K) - \overline{g}_{leak} (V - E_{leak}) \tag{1}$$

$$\frac{dm}{dt} = (1-m)\cdot\alpha_m(V) - m\cdot\beta_m(V) \quad (2)$$

$$\frac{dh}{dt} = (1-h)\cdot\alpha_h(V) - h\cdot\beta_h(V) \quad (3)$$

$$\frac{dn}{dt} = (1-n)\cdot\alpha_n(V) - n\cdot\beta_n(V) \quad (4)$$

The modeled change in the membrane's potential V depends on the applied current $I_e(t)$, conductances to specific ions on a piece of the membrane ($\overline{g}_{Na}$, $\overline{g}_K$, $\overline{g}_{leak}$), and the state of channels.

The included K and Na ion channels are proteins composed of 4 subunits, or gates, that can be either open or closed. Millions of these channels populate a modeled piece. The dynamical variables m, n, and h describe the fraction of open subunits across the channels on the membrane. Thus, the solutions of these variables can vary in the $[0, 1]$ range.

The functions $\alpha_{[m,n,h]}$ and $\beta_{[m,n,h]}$ (Appendix A) describe voltage-dependent transition rates: α describes the rate at which a fraction of closed gates opens, and β – the rate at which a fraction of opened gates shuts for a corresponding subunit. For the sodium and potassium channels, constants $\overline{g}_{Na}$ and $\overline{g}_K$ describe the maximal conductance to the corresponding ion when all the subunits are in the open state. The small non-voltage dependent conductance to Na and K ions is captured by $\overline{g}_{leak}$.

Numerical integrators for Hodgkin-Huxley model

A numerical solver estimates the trajectory of solution $x(t)$ of the initial value problem ($x(0) = x_0$) during a period T at a given grid:

$$\frac{dx(t)}{dt} = f(x,t), \qquad x(0) = x_0, \qquad t \in [0,T]. \quad (5)$$

2.2 Classical solvers

Classical numerical ODE solvers iteratively evaluate the vector field $f(x,t)$ at discrete times $t_1, \ldots, t_N$, providing single point estimates of solution with a global bound on error. The most frequently used integrator for the Hodgkin-Huxley model is Exponential Euler [18] with step size 10^{-2} ms (Appendix A, Table 1). This choice is is motivated by the observation that the set of Hodgkin-Huxley equations is a conditionally linear system of ODEs [7]. Assuming that the modeled variables have very little variation within a step $[t_n\ ,\ t_{n+1}]$, their value is set to a constant at every step. Then each variable appears linearly in the equation governing its time evolution and the ODE can be solved analytically over the time interval $\Delta t_{n+1} = t_{n+1} - t_n$ (detailed explanation in Appendix A).

2.3 Probabilistic solvers

Probabilistic solvers return a posterior distribution of estimated ODE solution

$$p(x(t)|x(t_0) = x_0, \{\dot{x}(t_n) = f(x(t_n), t_n)\}_{n=0}^N), \quad (6)$$

providing uncertainty estimates of the solution. To estimate the posterior, a filtering-based probabilistic solver formulates the inference of solution as a regression problem in a Bayesian state-space model [11, 13].

To estimate the posterior over solution, a probabilistic solver depends on the prior and likelihood, or measurement, models.

Prior

The prior is modeled with a Gauss-Markov process that solves the linear, time-invariant stochastic differential equation (SDE) of the following form :

$$dX(t) = AX(t)dt + \kappa B dW(t), \tag{7}$$

where A is drift matrix, B is a diffusion matrix, $\kappa \in \mathbb{R}$ is a diffusion scaling parameter and $W(t)$ is a vector of a standard Wiener process.

The structure of drift and diffusion matrices determines the covariance of the prior process. In general form, for a d-dimensional ODE the drift $A \in \mathbb{R}^{d(q+1)\times d(q+1)}$ and diffusion $B \in \mathbb{R}^{d(q+1)\times d}$ matrices are

$$A = \begin{bmatrix} 0 & I_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_d \\ f_{q,0} & f_{q,1} & \cdots & f_{q,q} \end{bmatrix}, \qquad B = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma \end{pmatrix}, \tag{8}$$

where $f_{q,i}$ are the coefficients in the corresponding characteristic polynomial of A, parametrizing the covariance matrix of the prior [13].

Measurement model

The measurement model links the prior to the solution of the ODE, describing the difference between the first derivative of the prior, and the evaluation of the vector field at the solution :

$$Z(t) = E_1 X(t) - f(E_0 X(t), t) = 0, \tag{9}$$

where E_i is a selection matrix, such that $E_i X(t) = X^{(i)}(t)$, and f is the modeled ODE.

Approximate Gaussian Inference

By conditioning the prior on the samples from the measurements $Z(t) = 0$ from a discrete grid $\{t_n\}_{n=0}^N$, one can formulate the inference problem as :

$$X_0 \sim N(\mu_0, \Sigma_0) \tag{10}$$

$$X_{n+1}|X_n \sim N(\Phi(\Delta t_n)X_n, \kappa^2 Q(\Delta t_n)) \tag{11}$$

$$Z_n|X_n \sim N(E_1 X_n - f(E_0 X_n, t_n), R) \tag{12}$$

$$z_n = 0, \quad n = 1, \ldots, N, \tag{13}$$

where z_n are the samples from the measurement model Z_n, $\Delta t_n = t_{n+1} - t_n$ is the step-size of the $(n+1)$-th step.

The prior satisfies Gaussian initial condition $X(0)$ and linear Gaussian transition densities (Equation 11). The transition matrices $\Phi(\Delta t_n)$ and $Q(\Delta t_n)$ depend on the step size Δt_n and are known in closed form [19].

Since we want to solve the ODE exactly, the variance of measurements R is 0, resulting in Dirac distribution of the measurement model $Z_n|X_n \sim \delta(E_1 X_n - f(E_0 X_n, t_n))$.

Extended Kalman Filter

The solution of the resulting non-linear Gauss-Markov regression problem can be efficiently approximated by Kalman filters and smoothers. At every step of simulation they iterate the following predict and update steps:

Predict

$$\mu_{n+1}^P = A(\Delta t_n)\mu_n^F \tag{14}$$

$$\Sigma_{n+1}^P = A(\Delta t_n)\Sigma_n^F A^T(\Delta t_n) + Q(\Delta t_n) \tag{15}$$

where $h = t_{n+1} - t_n$ is the step size, μ_{n+1}^P and Σ_{n+1}^P are predictive mean and covariance, and the transition matrices $A(\Delta t_n)$ and $Q(\Delta t_n)$ are given in the prior (Equation 11).

Update

$$\widehat{z}_{n+1} = z(t_{n+1}, \mu_{n+1}^P) \tag{16}$$

$$S_{n+1} = H_{n+1}\Sigma_{n+1}^P H_{n+1}^T \tag{17}$$

$$K_{n+1} = \Sigma_{n+1}^P H_{n+1}^T S_{n+1}^{-1} \tag{18}$$

$$\mu_{n+1}^F = \mu_{n+1}^P + K_{n+1}(z_{n+1} - \widehat{z}_{n+1}) \tag{19}$$

$$\Sigma_{n+1}^F = \Sigma_{n+1}^P - K_{n+1}S_{n+1}K_{n+1}^T, \tag{20}$$

where $z(t_{n+1}, \mu_{n+1}^P) = E_1\mu_{n+1}^P - f(E_0\mu_{n+1}^P, t_{n+1})$ is the error between f at the estimated solution and the true solution's derivative. The sample from the measurement model z_{n+1} is the desired error (Equation 13). To deal with non-linear measurements in likelihood, first-order extended Kalman filter (EKF1) makes a first-order Taylor series expansion around filtering mean μ_n^F at each step of simulation, such that

$$H_n = E_1 - J_f(E_0\mu_n^F, t_n)E_0, \tag{21}$$

where J_f is the Jacobian of f. The obtained μ_{n+1}^F and Σ_{n+1}^F are called filtering mean and covariances.

2.4 Priors

In the state-space model, the state vector $X(t) = [X^{(0)}(t), X^{(1)}(t) \cdots X^{(q)}(t)]$ models the evolution of the solution $x(t) = X^{(0)}(t)$ and its q derivatives $(X^{(1)}(t) \cdots X^{(q)}(t))$.

Integrated Wiener process

The most common prior choice in the probabilistic ODE solver literature is the q-times integrated Wiener process (IWP) [13]. Its q-th derivative is modeled with a Wiener process and the $X^{(0)}(t), X^{(1)}(t) \cdots X^{(q-1)}(t)$ are corresponding integrals. The drift and diffusion matrices for the d-dimensional IWP are

$$A_{IWP} = \begin{bmatrix} 0 & I_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_d \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \qquad B_{IWP} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I_d \end{pmatrix}. \tag{22}$$

Matérn process

The model matrices for the Matérn process differ to the IWP in the last row of the drift matrix (Eq. 8): $f_{q,i} \neq 0, i = 0, \cdots q$. The corresponding Matérn covariance function depends on the smoothness parameter ν, and characteristic length scale [20]. In the experiment with Matérn prior, the smoothness parameter is set to $\nu = q - 0.5 = 3 - 0.5 = 2.5$, according to the existing implementation [21]. The characteristic length scale influences the variance and smoothness of the samples and their q derivatives (Figure 1).

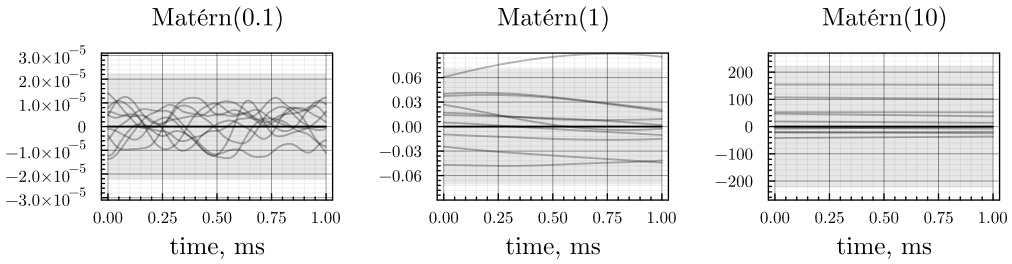


Figure 1: Influence of length scale parameter on Matérn prior. Extrapolation with Matérn prior during 1 s with $\mu = 0$. Left: Matérn(number of derivatives = 3, length scale = 0.1), middle: Matérn(number of derivatives = 3, length scale = 1), right: Matérn(number of derivatives = 3, length scale = 10).

Integrated Ornstein Uhlenbeck process

The SDE of the Ornstein Uhlenbeck process is

$$dX_t = \theta(X_t - \mu)dt + \sigma dW_t. \tag{23}$$

If the current value of the process X_t is smaller or bigger than the mean μ, the process tends to drift back to μ with strength θ. The magnitude of the rate parameter influences the probability of the process to stay outside of the mean. For negative θ, the process X_t stays close to the mean μ with high probability. When the rate is positive, the process diverges exponentially from the mean.

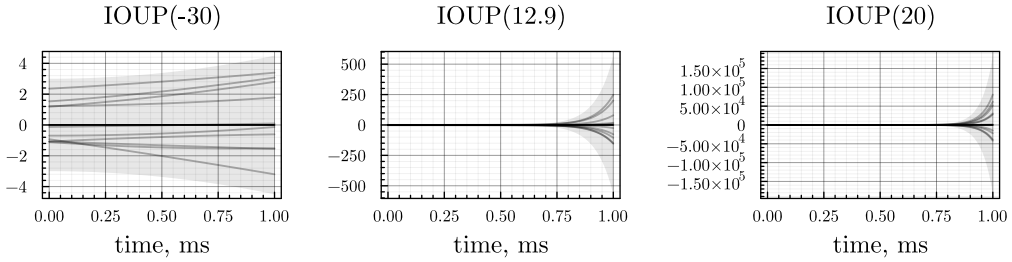


Figure 2: Influence of rate on IOUP prior. Extrapolation with IOUP prior during 1 s with $\mu = 0$. Left: IOUP(number of derivatives = 3, rate = -30), middle: IOUP(number of derivatives = 3, rate = 12.9), right: IOUP(number of derivatives = 3, rate = 20).

The prior of a probabilistic solver is modeled by a q-times integrated Ornstein Uhlenbeck process (IOUP), thus, given the notation in Equation 8, the rate and diffusion matrices are

$$A = \begin{bmatrix} 0 & I_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_d \\ 0 & 0 & \cdots & \theta \end{bmatrix}, \qquad B = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma \end{pmatrix}. \tag{24}$$

It is easy to see that for $\theta = 0$, the IOUP prion is equivalent to IWP.

2.5 Link functions

Link functions have been used for probabilistic solvers to match the support of prior and of the modeled variables [17]. In the Hodgkin-Huxley model, the opening probabilities of subunits can vary between 0 and 1, but the IWP prior can take values on the real line. Finding a new prior varying on $[0, 1]$ range corresponds to changing the dynamical model, such that its solution can take values on the real line.

A chosen link function easily adapting to various ranges is a scaled sigmoid, represented in general form by:

$$\sigma(x) = \frac{\alpha}{1 + e^{-\beta \cdot (x - x_0)}} + y_0, \tag{25}$$

where α sets the scale, β is the slope, y_0 is the y-offset, and x_0 is the x-offset.

For the illustration, the transformations are explained for the dynamics of channel m. The desired prior should take values on $[0, 1]$, thus, the scale α of corresponding sigmoid is 1, and the x- and y-offsets are 0. This link function maps $\mathbb{R} \to [0, 1]$, but to arrive at the transformed dynamical model, we also need to define its inverse transformation :

$$\sigma^{-1}(x) = \frac{-1}{\beta} \cdot log(\frac{\alpha}{y - y_0} - 1) + x_0. \tag{26}$$

Let's denote the dynamics of subunit m in the transformed $[-\infty, \infty]$ space as $\tilde{m}$. Given the ODE of the m-subunit: $\frac{dm}{dt} = f(m, t)$ (Equation 2), the following relationships hold:

$$\tilde{m}(t) = \sigma_m^{-1}(m(t)), \qquad m(t) = \sigma_m(\tilde{m}(t)). \tag{27}$$

Now, using chain rule, the ODE for $\tilde{m}$ can be written as:

$$\frac{d\tilde{m}}{dt} = \frac{d\tilde{m}}{dm} \cdot \frac{dm}{dt} = (\sigma_m^{-1})'(\sigma_m(\tilde{m}(t))) \cdot f(\sigma_m(\tilde{m}(t)), t). \tag{28}$$

The inference of solution with transformed prior corresponds to solving the ODE for $\tilde{m}$ with IWP prior

$$\tilde{m} \sim IWP(3), \tag{29}$$

and obtaining the solution with the link function

$$m(t) = \sigma_m(\tilde{m}(t)). \tag{30}$$

Similar transformations were done for channels n and h. The dynamics of channels n and h in the transformed space and their corresponding link functions are denoted by $\tilde{n}$, $\tilde{h}$ and σ_n, σ_h. Since each of subunits enters the ODE for voltage in non-transformed $[0, 1]$ range, the dynamics of voltage become:

$$C_M \frac{dV}{dt} = I_e(t) - \overline{g}_{Na} \boldsymbol{\sigma_m}(\tilde{\boldsymbol{m}})^3 \boldsymbol{\sigma_h}(\tilde{\boldsymbol{h}})(V - E_{Na}) - \overline{g}_K \boldsymbol{\sigma_n}(\tilde{\boldsymbol{n}})^4 (V - E_K) - \overline{g}_{leak}(V - E_{leak}). \tag{31}$$

These transformations do not require changes to the EKF1 algorithm, because they equivalent to linearizing another ODE in the update step (Equation 21).

Link functions for Hodgkin-Huxley ODE

The transformation for voltage bounded the values of prior within $[-110, 60]$ range with $\sigma_V(x)$:

$$\sigma_V(x) = \frac{170.0}{1 + e^{-0.05 \cdot x}} - 110.0 \tag{32}$$

The transformation for the m, n and h channels bounded the values within $[0, 1]$ range with $\sigma_{[m,n,h]}(x)$:

$$\sigma_{[m,n,h]}(x) = \frac{1}{1 + e^{-1.0 \cdot x}} \tag{33}$$

2.6 Practical considerations

Error

Since there is no closed-form solution, an output of a solver with very low error tolerance and, consequently, very small step size is considered a good proxy for the true solution. In the further analysis, the solution of Verner's 9/8 Runge-Kutta method [22] with absolute and relative tolerances $= 10^{-9}$ is regarded as the reference solution.

The accuracy of a solver's estimate was measured with trajectory root-mean-squared error (tRMSE) (Eq. 34), computed across four dimensions of the ODE (V, m, n, h) during the 50 ms of simulation ($N = \Delta t \cdot T$, where T is the duration of simulation and Δt is the step size). For a probabilistic solver, the estimate of solution at each step is the filtering mean $\widehat{y}_i = E_0 \mu_i^F$.

$$tRMSE = \sqrt{\frac{1}{N \cdot 4} \sum_{ch=1}^{4} \sum_{i=1}^{N} \|\widehat{y_i^{ch}} - y_i^{ch}\|^2} \tag{34}$$

Probabilistic solver parameters

The classical solvers for Hodgkin-Huxley typically use a step size of 10^{-2} ms (refer to Appendix A, Table 1). Therefore, before extending the analysis to multiple step sizes, I first evaluated the performance of a prior using this step size.

The error of a probabilistic solver is dependent on the number of derivatives in the prior [12, 19]. In order to isolate the effect of selecting a different SDE generating covariance for the prior, I only considered priors with $q = 3$ derivatives owing to their satisfactory performance (refer to Appendix A, Figure 13), and solvers with the Extended Kalman ODE Filter (EKF1) algorithm.

Therefore, the benchmark performance is the error made by the EKF1 solver with IWP(3) prior, with a fixed diffusion model and fixed step sizes. The performance of a solver is typically analyzed using work-precision diagrams at various step sizes, as illustrated in Figure 3.

3 Results

3.1 Choice of prior

Matérn process

In order to determine the optimal parametrization for the Matérn prior, I conducted a search on a grid ranging from 10^{-6} to 10^5 for the length scales parameter, utilizing a number of derivatives $q = 3$ and a smoothness value of $\nu = 2.5$.

For the step size $\Delta t = 0.01$ ms, the error of a solver with Matérn prior does not outperform the benchmark.

Integrated Ornstein Uhlenbeck process

To find the optimal parametrization of the IOUP, I evaluated tRMSE for different rates on a grid $[-10^4, 10^4]$. The optimal rate for step size $\Delta t = 0.01$ ms was positive $\theta = 13.9$ causing the variance to diverge faster from the mean.

To ensure that the optimal parametrization remains effective across different step sizes, the optimal rate was searched on the same grid for various step sizes. The grids were plotted for every step size

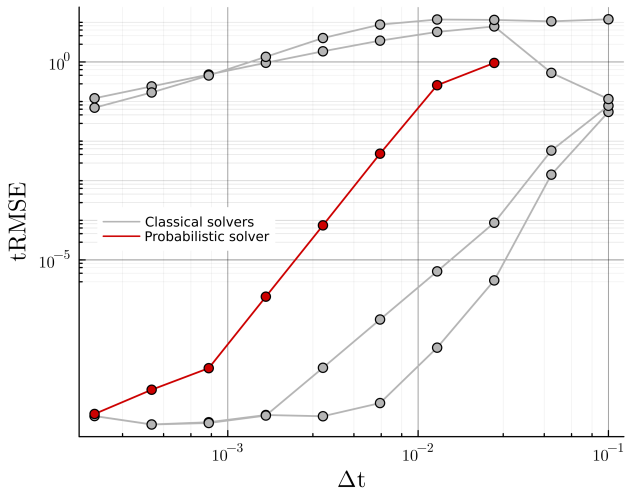


Figure 3: Probabilistic benchmark. The highlighted benchmark probabilistic solver has EKF1 algorithm and IWP(3) prior. The performance of classical solvers (in grey) sets a context for the probabilistic benchmark.

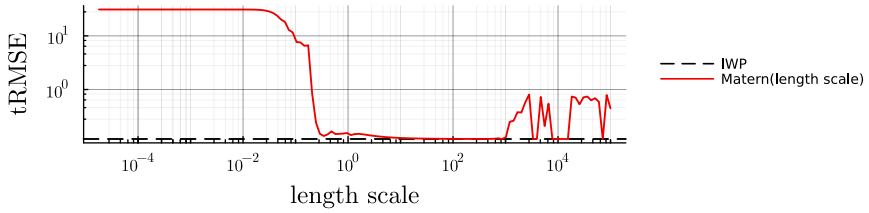


Figure 4: The impact of the length scale parameter in the Matérn(3, length scale) prior on the probabilistic solver's performance, in comparison to the benchmark IWP(3) prior. The step size is 10^{-2} ms.

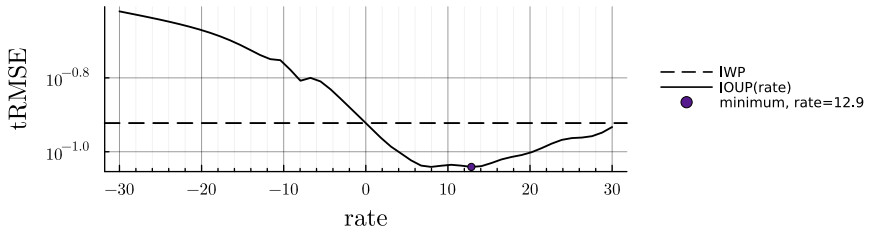


Figure 5: Effect of the rate in the IOUP(3, rate) prior on the tRMSE compared to the benchmark for a step size of $\Delta t = 10^{-2}$ ms. The grid range is focused within the $[-30, 30]$ interval.

to ensure that they covered the global minima. However, when scaled up for multiple step sizes, the decrease in error compared to the baseline was unnoticeable (Figure 6).

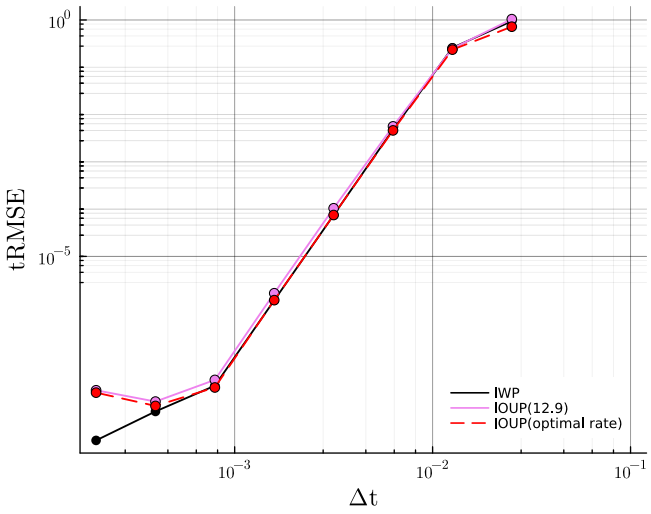


Figure 6: Impact of the optimal rate parameterization in the IOUP prior compared to the fixed rate in IOUP(number of derivatives = 3, rate = 12.9) and benchmark for various step sizes.

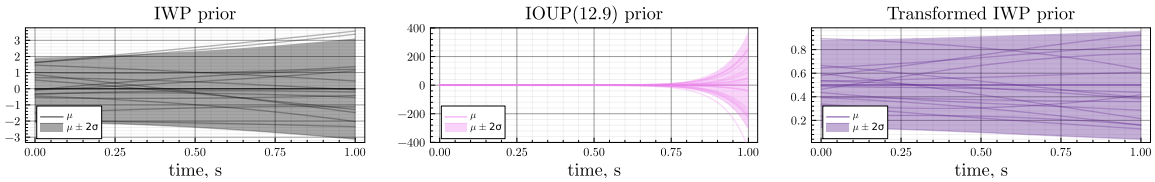


Figure 7: Extrapolation with different priors during 1 s. Left: IWP(number of derivatives = 3), middle: IOUP(number of derivatives = 3, rate = 12.9), transformed IWP(number of derivatives = 3)

3.2 Link functions

IWP prior

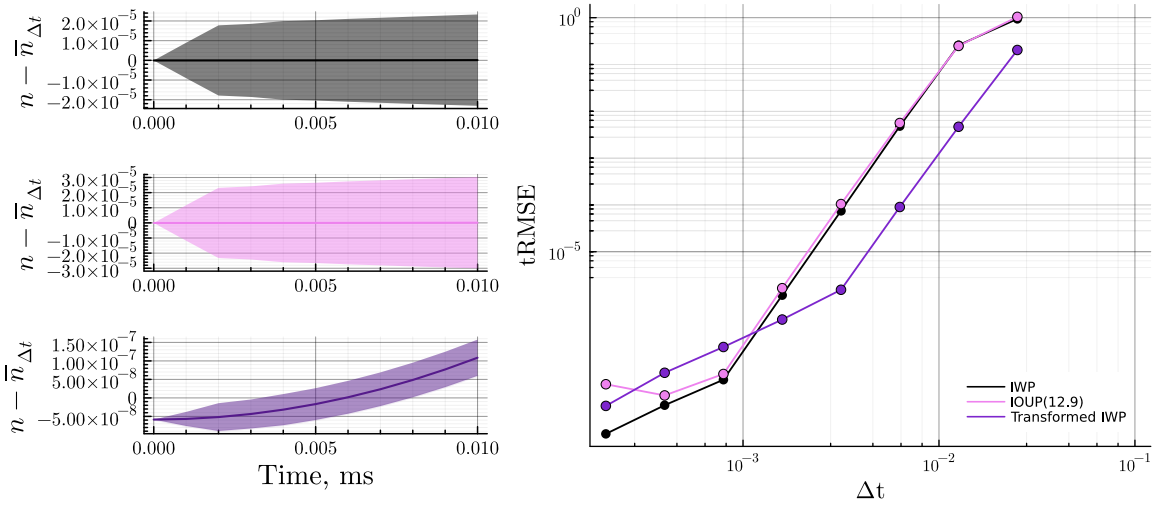
Restriction of parameter values with link functions resulted in a significant reduction in error across different step sizes. The equivalent results were seen with other transformation – scaled sine function (Appendix A, Figure 15). As seen in Figure 7, these transformations concentrated the probability mass in the desired range, while the variance of both the default IWP and IOUP priors increases, exceeding the $[0, 1]$ bounds.

The EKF1 uses the prior to extrapolate at each prediction step (Equation 14). Thus, the effect of a prior is influential within the extrapolation step. Figure 8 (a) shows this effect. The plotted mean and standard deviations are the extrapolations with different priors on the very first step of dynamics for channel n. The standard deviations are scaled with the square root of estimated diffusion. Since the dynamics are initialized with a non-zero value, I subtracted the mean over the interval $\overline{n}_{\Delta t}$ from the extrapolation trace.

On 1-step prediction with both IWP and IOUP priors, the standard deviation scales fast, while the prediction in the transformed space is highly concentrated around the mean.

IOUP and Matérn priors transformed with link functions

Using link functions led to a more substantial decrease in error compared to choosing a different SDE generating prior's covariance. Figures 9 and 10 demonstrate that selecting a different prior covariance



(a) Comparison of 1-step extrapolation with different priors.

(b) Influence of transformation on the error compared to the IOUP prior for multiple step sizes.

Figure 8: Influence of transformation on the error made by a probabilistic solver.
(a) Top: IWP(3), middle: IOUP(3, 12.9), bottom: transformed IWP(3). The plotted traces show the difference between the extrapolated value of the channel $n(t)$ during the initial step and the average value $\overline{n}_{\Delta t}$ over that step

does not notably reduce the error for the transformed ODE when compared with the baseline IWP(3).

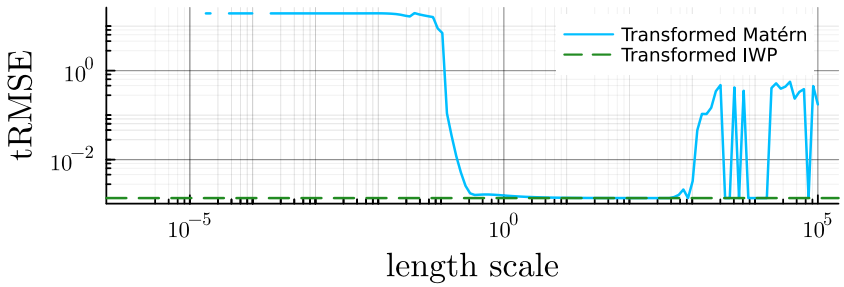


Figure 9: Influence of Matérn prior in the transformed ODE on the error for the step size $\Delta t = 10^{-2}$ ms.

Generalization to multiple number of derivatives in prior

Initially, I considered only the priors with number of derivatives equal to 3, because the convergence rates of priors with different number of derivatives are known [12, 19]. The question for the last set of experiments is whether the transformation of the ODE benefits the probabilistic solver across different number of derivatives in prior.

As can be seen on Figure 11, probabilistic solvers benefit from transformation across different number of derivatives in prior.

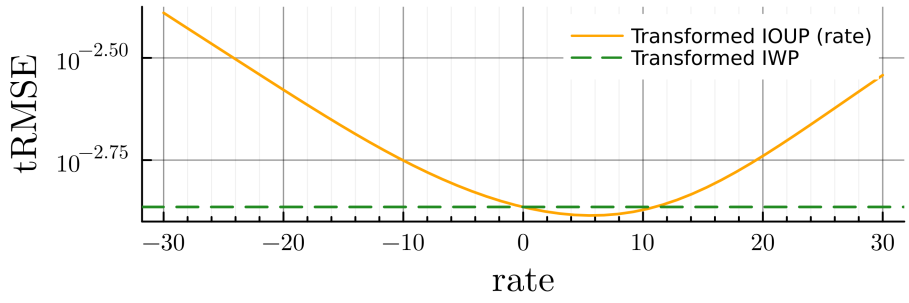


Figure 10: Influence of IOUP prior in the transformation on the error for the step size $\Delta t = 10^{-2}$ ms.

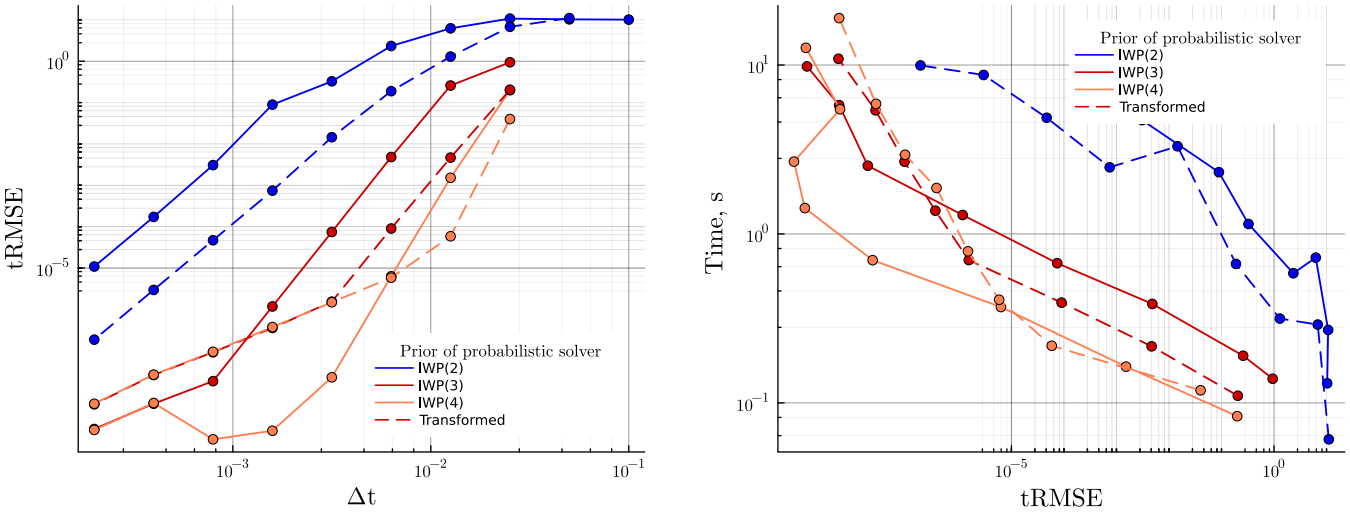


Figure 11: Impact of the transformation on priors with varying numbers of derivatives. Impact of the transformation on the error based on the step size (left) and runtime (right).

Influence of transformation on classical solvers

Since choosing a prior with a bounded variance equivalents to transforming an ODE with link functions, these transformations can also be used by classical solvers. The solution is then estimated by a classical solver in transformed space, and transformed to the original limits with link functions (Equation 27).

Although transformation of ODE reduced the error of probabilistic solver, classical solvers do not benefit from such transformation (Figure 12).

Implementation

All experiments are implemented in the Julia programming language [23] using ProbNumDiffEq.jl [21]. Reference solutions are computed with DifferentialEquations.jl [24]. Code for the implementation and experiments is publicly available on GitHub: https://github.com/sofiyagarkot/hh-julia.

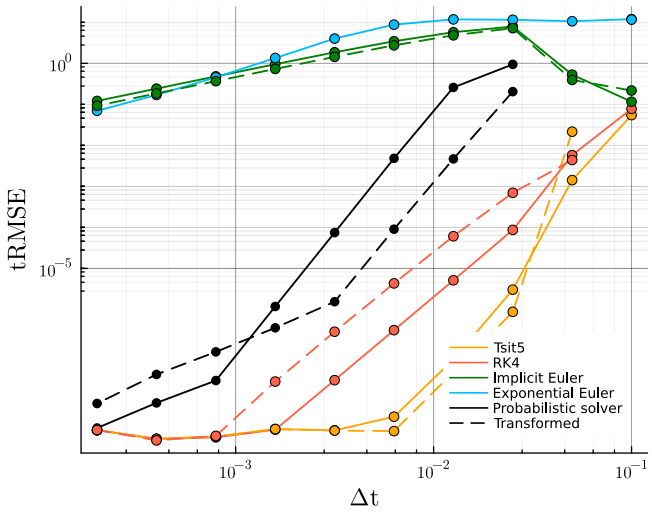


Figure 12: Influence of transformation on the error of classical solvers.

4 Discussion

We have shown that for the Hodgkin-Huxley ODE, the performance of a probabilistic solver does not significantly depend on the parametrization of covariance function, comparing Matérn or Integrated Ornstein Uhlenbeck processes in the prior. However, using link functions to get the prior with bounded variance benefits a probabilistic solver for the Hodgkin-Huxley model. The transformation of ODE with link functions significantly reduced the error of the probabilistic solver across different number of derivatives in prior, enabling to take bigger steps at a given error tolerance.

The examined parameterization of priors was limited exclusively to Matérn and IOUP priors, excluding other categories such as periodic or rational quadratic priors.

The performance of a probabilistic solver highly depends on the bounds of transformation, and it still needs to be determined which transformations are more suitable for the Hodgkin-Huxley model.

The experiments were limited to the fixed time stepping method and fixed diffusion models. One of the further directions could be to test whether probabilistic solvers with adaptive step size selection and dynamic diffusion models also benefit from transformation of priors.

Finally, there exist numerous ODEs that might benefit from adjusting probabilistic priors based on known solution bounds. One of the future directions involves assessing whether these findings remain applicable to other dynamical systems.

Acknowledgements

I am very grateful for the supervision of Nathanael Bosch and Jonas Beck, both of whom have greatly supported and taught me during the rotation! Additionally, I was surrounded by really supportive and knowledgeable lab mates in the Hennig group and the Berens lab, to whom I also would like to thank. I am grateful to Professors Berens and Hennig for the opportunity to participate in this interdisciplinary project at the intersection of their respective fields.

References

1. Hodgkin, A. L. & Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology* **117,** 500 (1952).
2. Bower, J. M., Beeman, D., Nelson, M. & Rinzel, J. The Hodgkin—Huxley Model. *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System,* 29–49 (1998).
3. Carnevale, N. T. & Hines, M. L. *The NEURON book* (Cambridge University Press, 2006).
4. Stimberg, M., Brette, R. & Goodman, D. F. Brian 2, an intuitive and efficient neural simulator. *eLife* **8** (eds Skinner, F. K., Calabrese, R. L., Skinner, F. K., Zeldenrust, F. & Gerkin, R. C.) e47314. ISSN: 2050-084X. `https://doi.org/10.7554/eLife.47314` (Aug. 2019).
5. Beeman, J. M. B. D. *The book of GENESIS* 2003.
6. Natschläger, T., Markram, H. & Maass, W. Computer models and analysis tools for neural microcircuits. *Neuroscience databases: a practical guide,* 123–138 (2003).
7. Chen, Z., Raman, B. & Stern, A. Structure-Preserving Numerical Integrators for Hodgkin–Huxley-Type Systems. *SIAM Journal on Scientific Computing* **42,** B273–B298. `https://doi.org/10.1137%2F18m123390x` (Jan. 2020).
8. Borgers, C. & Nectow, A. R. Exponential Time Differencing for Hodgkin–Huxley-like ODEs. *SIAM Journal on Scientific Computing* **35,** B623–B643 (2013).
9. Teymur, O., Zygalakis, K. & Calderhead, B. Probabilistic linear multistep methods. *Advances in Neural Information Processing Systems* **29** (2016).
10. Abdulle, A. & Garegnani, G. Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *Statistics and Computing* **30,** 907–932 (2020).
11. Tronarp, F., Kersting, H., Särkkä, S. & Hennig, P. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing* **29,** 1297–1315 (2019).
12. Tronarp, F., Särkkä, S. & Hennig, P. Bayesian ODE solvers: the maximum a posteriori estimate. *Statistics and Computing* **31,** 23 (2021).
13. Hennig, P., Osborne, M. A. & Kersting, H. P. *Probabilistic Numerics: Computation as Machine Learning* (Cambridge University Press, 2022).
14. Oesterle, J., Krämer, N., Hennig, P. & Berens, P. Probabilistic solvers enable a straight-forward exploration of numerical uncertainty in neuroscience models. *Journal of Computational Neuroscience* **50,** 485–503 (2022).
15. Bosch, N., Hennig, P. & Tronarp, F. Probabilistic Exponential Integrators. *Advances in Neural Information Processing Systems* **36** (2024).
16. Magnani, E., Kersting, H., Schober, M. & Hennig, P. Bayesian filtering for ODEs with bounded derivatives. *arXiv preprint arXiv:1709.08471* (2017).
17. Schmidt, J., Krämer, N. & Hennig, P. A probabilistic state space model for joint inference from differential equations and data. *Advances in Neural Information Processing Systems* **34,** 12374–12385 (2021).
18. Hairer, E. & Ch, L. Numerical solution of ordinary differential equations. *The Princeton companion to applied mathematics,* 293–305 (2012).
19. Kersting, H., Sullivan, T. J. & Hennig, P. Convergence rates of Gaussian ODE filters. *Statistics and computing* **30,** 1791–1816 (2020).

20. Särkkä, S. & Solin, A. *Applied stochastic differential equations* (Cambridge University Press, 2019).

21. Bosch, N. *ProbNumDiffEq.jl* `https://github.com/nathanaelbosch/ProbNumDiffEq.jl`.

22. Verner, J. H. Numerically optimal Runge–Kutta pairs with interpolants. *Numerical Algorithms* **53,** 383–396 (2010).

23. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: A Fresh Approach to Numerical Computing. *SIAM Review* **59,** 65–98. `https://doi.org/10.1137/141000671` (2017).

24. Rackauckas, C. & Nie, Q. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open Research Software* **5** (May 2017).

A Appendix

Classical solvers

Reference	Method	Δt, ms
CSIM [6]	Exponential Euler	0.001
NEURON [3]	Backward Euler	0.025
GENESIS [5]	Exponential Euler	0.01
BRIAN2 [4]	Exponential Euler	0.01
Chen et al. [7]	Splitting & Composition methods	0.1, 0.4

Table 1: Choices of step size in existing simulators and methods .

Semi-linearity

A conditionally linear system of ODEs is a system of the form

$$\dot{x_i} = a_i(x)x_i + b_i(x), \tag{35}$$

where the coefficients $a_i(x)$ and $b_i(x)$ are real-valued functions that depend on $x_j, j \neq i$ [7].

In the Equation 1 the dynamical variables m, n and h are enter the dynamics for voltage continuously. If the values of the variables are constant within a step:

$$m(t_n) = m_n, n(t_n) = n_n, h(t_n) = h_n, V(t_n) = V_n, \tag{36}$$

the ODE can be re-written as:

$$\dot{V} = \overbrace{\frac{1}{C_M}[-\overline{g_{Na}}m_n^3 n_n - \overline{g_K}n_n^4 - \overline{g_{leak}}]}^{\text{const} = c_{11}} V + \overbrace{\frac{1}{C_M}[I_e(t) + \overline{g_{Na}}m_n^3 h_n E_{Na} + \overline{g_K}n_n^4 E_K + \overline{g_{leak}}E_{leak}]}^{\text{const}=c_{12}} \tag{37}$$

$$\begin{aligned}
\dot{m} &= \overbrace{[-\alpha_m(V_n) - \beta_m(V_n)]}^{\text{const}=c_{21}} m + \overbrace{\alpha_m(V_n)}^{\text{const}=c_{22}} \\
\dot{h} &= \overbrace{-[\alpha_h(V_n) - \beta_h(V_n)]}^{\text{const}=c_{31}} h + \overbrace{\alpha_h(V_n)}^{\text{const}=c_{32}} \\
\dot{n} &= \overbrace{-[\alpha_n(V_n) - \beta_n(V_n)]}^{\text{const}=c_{41}} n + \overbrace{\alpha_n(V_n)}^{\text{const}=c_{42}} .
\end{aligned}$$

Thus, with the values of variables fixed, Exponential Euler method can solve the system analytically over the time interval of duration $t_{n+1} - t_n$. This allows both to deal with stiffness and take larger step sizes effectively.

Hodgkin-Huxley model

Additional functions

$$\alpha_m(V) = \frac{-0.32 \cdot (V - V_T - 13)}{e^{\frac{-(V-V_T-13)}{4}} - 1} \tag{38}$$

$$\alpha_h(V) = 0.128 \cdot e^{\frac{-(V-V_T-17)}{18}} \tag{39}$$

$$\alpha_n(V) = \frac{-0.032 \cdot (V - V_T - 15)}{e^{\frac{-(V-V_T-15)}{5}} - 1} \tag{40}$$

$$\beta_m(V) = \frac{0.28 \cdot (V - V_T - 40)}{e^{\frac{V-V_T-40}{5}} - 1} \tag{41}$$

$$\beta_h(V) = \frac{4}{1 + e^{\frac{-(V-V_T-40)}{5}}} \tag{42}$$

$$\beta_n(V) = 0.5 \cdot e^{\frac{-(V-V_T-10)}{40}} \tag{43}$$

Hodgkin-Huxley model parameters

The Hodgkin-Huxley model has a number of hyperparameters that have been set to the following values. The input current stimulation $I_e(t)$ was set to 500 μA during the whole period of simulation (50 ms). In the beginning of simulation the membrane is slightly depolarized $V_0 = -70\ mV$ with resting potential of membrane $V_T = -60\ mV$. The conductances were set to $\overline{g}_{Na} = 20.0\ mS$, $\overline{g}_K = 15.0\ mS$, $\overline{g}_{leak} = 0.1\ mS$, the equilibrium potentials for the ions were set to : $E_{Na} = 53\ mV$, $E_K = -107\ mV$, $E_{leak} = -70\ mV$. Membrane's capacitance was set to $C_m = 1.5\ \mu F$.

Probabilistic solvers

The error of a probabilistic solver depends on the number of derivatives in prior. The global convergence rates of a probabilistic ODE solver with IWP(q) prior are $O(h^q)$ [19].

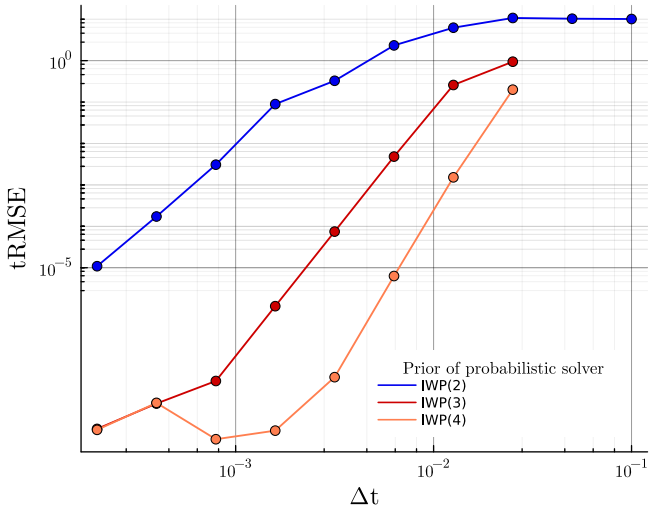


Figure 13: Influence of the number of derivatives in the prior of probabilistic solver on the accuracy of solution.

Voltage range

The range of values for voltage for Hodgkin-Huxley ODE is bounded within $[-82, +33]$ mV. Therefore, the dynamics of voltage were initially constrained to the $[-82, +33]$ mV range through a scaled sigmoid transformation (Eq. 44).

$$\tilde{\sigma}_V(x) = \frac{115.0}{1 + e^{-0.05 \cdot x}} - 82.0 \tag{44}$$

Surprisingly, the strict bounds on voltage resulted in worse performance compared the looser bound of $[-110.0, +60.0]$ mV (Eq. 32, Figure 14).

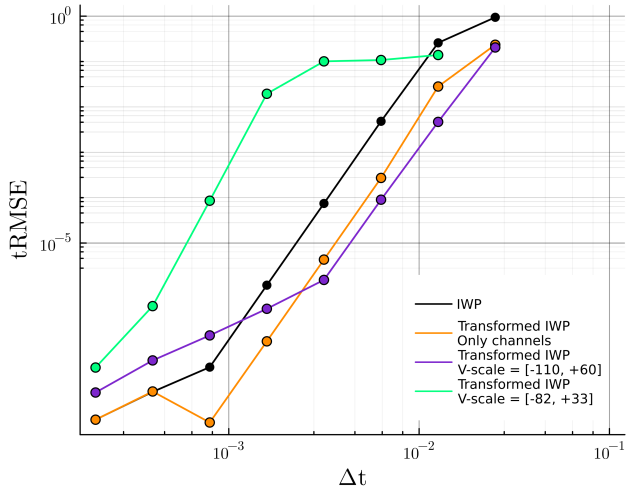


Figure 14: Influence of different ranges of voltage on the error.

Different transformations

$$sin_{[m,n,h]}(x) = 0.5sin(0.001x) + 0.5 \tag{45}$$

$$sin_V(x) = 170 \cdot sin(0.0001x) - 60 \tag{46}$$

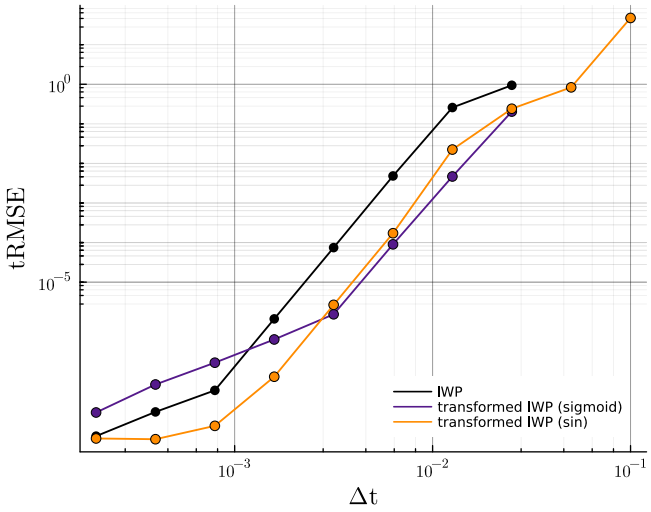


Figure 15: Influence of different transformation on the performance of a probabilistic solver. Comparing the link functions with sine transformation (orange) with to sigmoid (purple).