Supplementary Material

Manuscript: A Bayesian Physics-Informed Approach for Inverse Problems in Heat Transfer and Fluid Mechanics
Journal: Applied Mathematical Modelling
Authors: R.C. Fernandez, C.T.P. Zanini, A.M. Schmidt, H.S. Migon, A.J. Silva Neto
Corresponding author: rafaelc@dme.ufrj.br

Contents

1 Computational methods for Bayesian Inference

1.1 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) extends traditional Metropolis-Hastings by incorporating geometric information about the target distribution, leading to a more efficient exploration of the parameter space (1). The algorithm introduces auxiliary momentum variables $\mathbf{p}$ and defines a Hamiltonian system with potential energy $U(\boldsymbol{\theta}) = -\log \mathbb{P}(\boldsymbol{\theta}|\mathbf{y})$ and kinetic energy $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^\top \mathbf{M}^{-1}\mathbf{p}$, where $\mathbf{M}$ is a mass matrix. The total energy of the system, called the Hamiltonian system, is:

$$H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + K(\mathbf{p})$$

The system evolves according to Hamilton's equations:

$$\begin{aligned}
\frac{d\boldsymbol{\theta}}{dt} &= \nabla_{\mathbf{p}} H(\boldsymbol{\theta}, \mathbf{p}) = \mathbf{M}^{-1}\mathbf{p} \\
\frac{d\mathbf{p}}{dt} &= -\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}, \mathbf{p}) = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})
\end{aligned}$$

HMC generates proposals by numerically integrating these equations using the leapfrog integrator (1) over $L \in \mathbb{N}$ steps with step size $\epsilon > 0$:

$$\begin{aligned}
\mathbf{p}(t + \frac{1}{2}) &= \mathbf{p}(t) - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}(t)) \\
\boldsymbol{\theta}(t+1) &= \boldsymbol{\theta}(t) + \epsilon \mathbf{M}^{-1}\mathbf{p}(t + \frac{1}{2}) \\
\mathbf{p}(t+1) &= \mathbf{p}(t + \frac{1}{2}) - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}(t+1))
\end{aligned}$$

with $t = 0, \ldots, L-1$ representing the iterations of the leapfrog algorithm. The proposal is then accepted with probability:

$$\min\{1, \exp(H(\boldsymbol{\theta}, \mathbf{p}) - H(\boldsymbol{\theta}^*, \mathbf{p}^*))\}$$

where $(\boldsymbol{\theta}^*, \mathbf{p}^*)$ represents the proposed state after L iterations of the leapfrog algorithm, that is, $\boldsymbol{\theta}^* = \boldsymbol{\theta}(L+1)$ and $\mathbf{p}^* = \mathbf{p}(L)$. The NUTS adaptively tunes both L and ϵ during warmup, eliminating the need to manually specify these parameters while maintaining detailed balance. In our implementation, we use Stan, which employs NUTS as its primary sampler.

1.2 Variational Inference

Variational inference (VI) methods search among a predefined variational family $\mathcal{Q} = \{q_{\boldsymbol{\nu}}(\boldsymbol{\theta}) : \boldsymbol{\nu} \in \mathcal{V}\}$ (where its members are densities indexed by the variational parameter $\boldsymbol{\nu}$), the density $q_{\boldsymbol{\nu}^*}(\boldsymbol{\theta})$ that best approximates the posterior $\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{y})$ in terms of Kullback-Leibler divergence

$$\begin{aligned}
KL\left(q_{\boldsymbol{\nu}}(\boldsymbol{\theta}) \,||\, \mathbb{P}(\boldsymbol{\theta} \mid \mathbf{y})\right) :&= \mathbb{E}_{q_{\boldsymbol{\nu}}}[\log q_{\boldsymbol{\nu}}(\boldsymbol{\theta}) - \log \mathbb{P}(\boldsymbol{\theta}|\mathbf{y})] \\
&= \int [\log q_{\boldsymbol{\nu}}(\boldsymbol{\theta}) - \log \mathbb{P}(\boldsymbol{\theta}|\mathbf{y})] q_{\boldsymbol{\nu}}(\boldsymbol{\theta}) d\boldsymbol{\theta}.
\end{aligned}$$

The objective of variational inference is therefore to determine $\widehat{\boldsymbol{\nu}} = \arg\min_{\boldsymbol{\nu}\in\mathcal{V}} KL\left(q_{\boldsymbol{\nu}}(\boldsymbol{\theta})|\,|\,\mathbb{P}(\boldsymbol{\theta}\mid\mathbf{y})\right)$, which is equivalent to maximizing the evidence lower bound (ELBO)

$$ELBO(\mathbf{y},\boldsymbol{\nu}) := \mathbb{E}_{q_{\boldsymbol{\nu}}}[\log\mathbb{P}(\mathbf{y}\mid\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\nu}}(\boldsymbol{\theta})]$$

over the variational parameters $\boldsymbol{\nu}\in\mathcal{V}$. In fact, this equivalence is a direct consequence of the following identity $KL\left(q_{\boldsymbol{\nu}}(\boldsymbol{\theta})\,||\,\mathbb{P}(\boldsymbol{\theta}\mid\mathbf{y})\right) = -ELBO(\mathbf{y},\boldsymbol{\nu}) + \log\mathbb{P}(\mathbf{y})$.

Many classes of algorithms for variational inference have been proposed in the literature. The coordinate ascending variational inference (CAVI) is based on the mean-field assumption in which the joint variational distribution of the parameters is the product of its marginals: $q_{\boldsymbol{\nu}}(\boldsymbol{\theta}) = \prod_{j=1}^{K} q_{\boldsymbol{\nu}_j}(\boldsymbol{\theta}_j)$ for $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_K)$ (**?**). This assumption generally leads to fast convergence at the cost of the often oversimplifying assumption of independence in the blocks of parameters $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_K$. Other algorithms for VI such as (**?**) allow for more flexible variational families that do not need to rely on mean-field assumptions. They rely on stochastic gradient ascent-based algorithms that rewrite the ELBO so that the gradient operator can change order with the expectation, so the result $\nabla_{\boldsymbol{\nu}} ELBO(\mathbf{y},\boldsymbol{\nu})$ is an expected value with respect to the variational distribution. The resulting expectation can then be estimated via Monte Carlo methods. In this paper, we will focus on the Automatic Differentiation Variational Inference (ADVI) approach of (2) since it provides a flexible variational family with good scalability for large datasets and seamless implementation in Stan.

More specifically, suppose that the original parameters $\boldsymbol{\theta}$ vary in a subset $\Lambda\subset\mathbb{R}^d$ with $\Lambda\neq\mathbb{R}^d$. This happens, for example, when one or more entries of $\boldsymbol{\theta}$ lie in constrained subsets of $\mathbb{R}$, for example $\boldsymbol{\theta} = \mathbb{R}^d\times\mathbb{R}^+$ in the case of the observational variance $\sigma_y^2\in\mathbb{R}^+$. ADVI considers the vector of transformed parameters $\boldsymbol{\xi} = T(\boldsymbol{\theta})$, where $T:\Lambda\to\mathbb{R}^d$ is a differentiable and invertible transformation such that $\boldsymbol{\xi}$ lies in $\mathbb{R}^d$, without restrictions. To avoid the Mean-Field assumption, ADVI specifies a multivariate normal variational family on $\boldsymbol{\xi}\sim N(\mathbf{m},\boldsymbol{\Sigma})$ with a full covariance matrix $\boldsymbol{\Sigma}$, which implies a full dependence structure on the original parameters $\boldsymbol{\theta} = T^{-1}(\boldsymbol{\xi})$.

The reparameterization $\boldsymbol{\xi} = \mathbf{m} + \mathbf{L}\,\boldsymbol{\epsilon},\ \boldsymbol{\epsilon}\sim N(\mathbf{0},\mathbf{I})$ enables calculation of the gradient of the ELBO as an expectation, which can be approximated via Monte Carlo:

$$\begin{aligned}
\nabla_{\boldsymbol{\nu}} ELBO(\mathbf{y},\boldsymbol{\nu}) &= \nabla_{\boldsymbol{\nu}}\mathbb{E}_{q_{\boldsymbol{\nu}}(\boldsymbol{\xi})}\left[\log\widetilde{p}(\mathbf{y},\boldsymbol{\xi}) - \log q_{\boldsymbol{\nu}}(\boldsymbol{\xi})\right] \\
&= \nabla_{\boldsymbol{\nu}}\mathbb{E}_{\boldsymbol{\epsilon}\sim N(\mathbf{0},\mathbf{I})}\left[\log\mathbb{P}(\mathbf{y},T^{-1}(\boldsymbol{\xi})) + \log|J(\boldsymbol{\xi})| - \log N(\boldsymbol{\xi};\ \mathbf{m},\mathbf{L}\mathbf{L}^\top)\Big|_{\boldsymbol{\xi}=\mathbf{m}+\mathbf{L}\boldsymbol{\epsilon}}\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim N(\mathbf{0},\mathbf{I})}\left\{\nabla_{\boldsymbol{\nu}}\left[\log\mathbb{P}(\mathbf{y},T^{-1}(\boldsymbol{\xi})) + \log|J(\boldsymbol{\xi})| - \log N(\boldsymbol{\xi};\ \mathbf{m},\mathbf{L}\mathbf{L}^\top)\Big|_{\boldsymbol{\xi}=\mathbf{m}+\mathbf{L}\boldsymbol{\epsilon}}\right]\right\} \\
&\approx \frac{1}{M}\sum_{\ell=1}^{M}\nabla_{\boldsymbol{\nu}}\left[\log\mathbb{P}(\mathbf{y},T^{-1}(\boldsymbol{\xi})) + \log|J(\boldsymbol{\xi})| - \log N(\boldsymbol{\xi};\ \mathbf{m},\mathbf{L}\mathbf{L}^\top)\Big|_{\boldsymbol{\xi}=\mathbf{m}+\mathbf{L}\boldsymbol{\epsilon}^{(\ell)}}\right]
\end{aligned}$$

where $\boldsymbol{\epsilon}^{(\ell)}\overset{\text{iid}}{\sim} N(\mathbf{0},\mathbf{I})$, $\ell = 1,\ldots,M$, are Gaussian random noise vectors, $N(;\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the density function of a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at , $|J(\boldsymbol{\xi})|$ denotes the determinant of Jacobian transformation $\boldsymbol{\xi}\mapsto T^{-1}(\boldsymbol{\xi})$ and $\widetilde{p}(\mathbf{y},\boldsymbol{\xi}) = \mathbb{P}(\mathbf{y},\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=T^{-1}(\boldsymbol{\xi})}\times|J(\boldsymbol{\xi})|$. More details, including complete derivations of the ADVI algorithms, can be found, for example, in (**?**).

1.3 Stan implementation

Manual execution of a MCMC algorithm can be arduous, even for relatively straightforward scenarios. Figure 1 provides the actual code related to the implementation of the exponential growth model within Stan.

```
data {
  int<lower=0> n; // sample size
  vector[n] t; // time
  vector[n] y; // response variable
}
parameters {
  real<lower=0> lambda; // growth rate
  real<lower=0> sigma; // standard deviation
}
model {
  for (i in 1:n) {
    y[i] ~ normal(exp(lambda * t[i]), sigma); // likelihood function
  }
  lambda ~ normal(0, 10); // prior for lambda
  sigma ~ cauchy(0, 10); // prior for sigma
}
```

Figure 1: Bayesian exponential growth model coded in Stan.

2 Exponential Growth B-PIGP

This appendix presents the mathematical derivation of the B-PIGP for the exponential growth equation. We derive all covariance components and conditioning formulas needed for the Bayesian inference framework.

2.1 Problem Formulation

For the exponential growth problem, the ODE system is given by:

$$\frac{du(t)}{dt} - \lambda u(t) = 0, \quad t \in [0, T] \tag{1}$$

$$u(0) = u_0. \tag{2}$$

Therefore, the corresponding linear differential operator $\mathcal{L}_{\boldsymbol{\lambda}}$ is:

$$\mathcal{L}_{\boldsymbol{\lambda}} = \frac{d}{dt} - \lambda. \tag{3}$$

2.2 Gaussian Process Prior

We define a Gaussian Process prior for the solution $u(t)$ with mean function $\mu_{\boldsymbol{\beta}}(t)$:

$$u(t) \sim \mathcal{GP}(\mu_{\boldsymbol{\beta}}(t), k_{\boldsymbol{\psi}}(t, t')). \tag{4}$$

For simplicity, we set:

$$\mu_{\boldsymbol{\beta}}(t) = 0. \tag{5}$$

The covariance function is the squared exponential kernel:

$$k_{\boldsymbol{\psi}}(t, t') = \sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right), \tag{6}$$

where $\boldsymbol{\psi} = (\sigma_k^2, r)$ represents the kernel hyperparameters.

2.3 Joint Process Distribution

The joint Gaussian process for the solution and its differential operator is:

$$\begin{bmatrix} u(t) \\ \mathcal{L}_{\boldsymbol{\lambda}} u(t) \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathrm{Cov}(u,u) & \mathrm{Cov}(u, \mathcal{L}_{\boldsymbol{\lambda}} u) \\ \mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}} u, u) & \mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}} u, \mathcal{L}_{\boldsymbol{\lambda}} u) \end{bmatrix}\right). \tag{7}$$

2.4 Covariance Components

We now derive the covariance components between the function u and its differential operator $\mathcal{L}_{\boldsymbol{\lambda}} u$.

2.4.1 Function-Function Covariance

$$\mathrm{Cov}(u(t), u(t')) = \sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right) \tag{8}$$

2.4.2 Function-Operator Covariance

$$\begin{aligned} \mathrm{Cov}(u(t), \mathcal{L}_{\boldsymbol{\lambda}} u(t')) &= \mathrm{Cov}\left(u(t), \frac{du(t')}{dt'} - \lambda u(t')\right) \\ &= \frac{\partial}{\partial t'} \mathrm{Cov}(u(t), u(t')) - \lambda \,\mathrm{Cov}(u(t), u(t')) \qquad (9) \\ &= \frac{\partial}{\partial t'} k_{\boldsymbol{\psi}}(t, t') - \lambda k_{\boldsymbol{\psi}}(t, t') \qquad (10) \end{aligned}$$

Computing the kernel derivative:

$$\begin{aligned} \frac{\partial}{\partial t'} k_{\boldsymbol{\psi}}(t, t') &= \frac{\partial}{\partial t'}\left[\sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right)\right] \\ &= \sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right) \times \frac{\partial}{\partial t'}\left[-\frac{(t-t')^2}{2r^2}\right] \\ &= \sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right) \times \frac{t-t'}{r^2} \\ &= k_{\boldsymbol{\psi}}(t, t') \times \frac{t-t'}{r^2}. \qquad (11) \end{aligned}$$

Combining terms in equation (9):

$$\begin{aligned} \mathrm{Cov}(u(t), \mathcal{L}_{\boldsymbol{\lambda}} u(t')) &= k_{\boldsymbol{\psi}}(t, t') \frac{t-t'}{r^2} - \lambda k_{\boldsymbol{\psi}}(t, t') \\ &= k_{\boldsymbol{\psi}}(t, t') \left(\frac{t-t'}{r^2} - \lambda\right). \qquad (12) \end{aligned}$$

2.4.3 Operator-Function Covariance

$$\begin{aligned}\operatorname{Cov}(\mathcal{L}_{\boldsymbol{\lambda}} u(t), u(t')) &= \operatorname{Cov}\left(\frac{du(t)}{dt} - \lambda u(t), u(t')\right) \\ &= \frac{\partial}{\partial t}\operatorname{Cov}(u(t), u(t')) - \lambda \operatorname{Cov}(u(t), u(t')) \quad (13) \\ &= \frac{\partial}{\partial t} k_{\boldsymbol{\psi}}(t, t') - \lambda k_{\boldsymbol{\psi}}(t, t'). \quad (14)\end{aligned}$$

Computing the kernel derivative:

$$\begin{aligned}\frac{\partial}{\partial t} k_{\boldsymbol{\psi}}(t, t') &= \frac{\partial}{\partial t}\left[\sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right)\right] \\ &= \sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right) \times \frac{\partial}{\partial t}\left[-\frac{(t-t')^2}{2r^2}\right] \\ &= \sigma_k^2 \exp\left(-\frac{(t-t')^2}{2r^2}\right) \times \left[-\frac{t-t'}{r^2}\right] \\ &= -k_{\boldsymbol{\psi}}(t, t') \times \frac{t-t'}{r^2}. \quad (15)\end{aligned}$$

Combining terms in equation (13):

$$\begin{aligned}\operatorname{Cov}(\mathcal{L}_{\boldsymbol{\lambda}} u(t), u(t')) &= -k_{\boldsymbol{\psi}}(t, t')\frac{t-t'}{r^2} - \lambda k_{\boldsymbol{\psi}}(t, t') \\ &= k_{\boldsymbol{\psi}}(t, t')\left(-\frac{t-t'}{r^2} - \lambda\right). \quad (16)\end{aligned}$$

2.4.4 Operator-Operator Covariance

$$\begin{aligned}&\operatorname{Cov}(\mathcal{L}_{\boldsymbol{\lambda}} u(t), \mathcal{L}_{\boldsymbol{\lambda}} u(t')) \\ &\quad = \operatorname{Cov}\left(\frac{du(t)}{dt} - \lambda u(t), \frac{du(t')}{dt'} - \lambda u(t')\right). \quad (17)\end{aligned}$$

Expanding equation (17):

$$= \operatorname{Cov}\left(\frac{du(t)}{dt}, \frac{du(t')}{dt'}\right) - \lambda \operatorname{Cov}\left(\frac{du(t)}{dt}, u(t')\right) - \lambda \operatorname{Cov}\left(u(t), \frac{du(t')}{dt'}\right) + \lambda^2 \operatorname{Cov}(u(t), u(t')).$$

In terms of kernel derivatives:

$$= \frac{\partial^2}{\partial t \partial t'} k_{\boldsymbol{\psi}}(t, t') - \lambda \frac{\partial}{\partial t} k_{\boldsymbol{\psi}}(t, t') - \lambda \frac{\partial}{\partial t'} k_{\boldsymbol{\psi}}(t, t') + \lambda^2 k_{\boldsymbol{\psi}}(t, t'). \quad (18)$$

Computing the second derivative:

$$\begin{aligned}\frac{\partial^2}{\partial t \partial t'} k_{\boldsymbol{\psi}}(t, t') &= \frac{\partial}{\partial t}\left(k_{\boldsymbol{\psi}}(t, t')\frac{t-t'}{r^2}\right) \\ &= \frac{\partial k_{\boldsymbol{\psi}}(t, t')}{\partial t}\frac{t-t'}{r^2} + k_{\boldsymbol{\psi}}(t, t')\frac{\partial}{\partial t}\left(\frac{t-t'}{r^2}\right) \\ &= -k_{\boldsymbol{\psi}}(t, t')\frac{t-t'}{r^2}\frac{t-t'}{r^2} + k_{\boldsymbol{\psi}}(t, t')\frac{1}{r^2} \\ &= k_{\boldsymbol{\psi}}(t, t')\left(\frac{1}{r^2} - \frac{(t-t')^2}{r^4}\right). \quad (19)\end{aligned}$$

Combining all terms:

$$\begin{aligned}\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u(t), \mathcal{L}_{\boldsymbol{\lambda}}u(t')) = k_{\boldsymbol{\psi}}(t,t')\left(\frac{1}{r^2} - \frac{(t-t')^2}{r^4}\right) \\ + \lambda k_{\boldsymbol{\psi}}(t,t')\frac{t-t'}{r^2} \\ + \lambda k_{\boldsymbol{\psi}}(t,t')\frac{t'-t}{r^2} \\ + \lambda^2 k_{\boldsymbol{\psi}}(t,t').\end{aligned}$$

Simplifying:

$$\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u(t), \mathcal{L}_{\boldsymbol{\lambda}}u(t')) = k_{\boldsymbol{\psi}}(t,t')\left[\frac{1-(t-t')^2/r^2}{r^2} + \lambda^2\right]. \tag{20}$$

2.5 Conditional Distribution

The conditional distribution incorporating the ODE constraint is:

$$u(t) \mid \{\mathcal{L}_{\boldsymbol{\lambda}}u(t) = 0, \boldsymbol{\lambda}, \sigma_y^2\} \sim \mathcal{GP}(\boldsymbol{\mu}_{u|\mathcal{L}}, \boldsymbol{\Sigma}_{u|\mathcal{L}}). \tag{21}$$

With conditional parameters:

$$\boldsymbol{\mu}_{u|\mathcal{L}} = \mu_{\boldsymbol{\beta}}(t) + \mathrm{Cov}(u, \mathcal{L}_{\boldsymbol{\lambda}}u)[\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u, \mathcal{L}_{\boldsymbol{\lambda}}u)]^{-1}(0 - \mathcal{L}_{\boldsymbol{\lambda}}\mu_{\boldsymbol{\beta}}(t)), \tag{22}$$

$$\boldsymbol{\Sigma}_{u|\mathcal{L}} = \mathrm{Cov}(u,u) - \mathrm{Cov}(u, \mathcal{L}_{\boldsymbol{\lambda}}u)[\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u, \mathcal{L}_{\boldsymbol{\lambda}}u)]^{-1}\,\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u, u). \tag{23}$$

Measurement error is incorporated by:

$$\boldsymbol{\Sigma}_{y|\mathcal{L}} = \boldsymbol{\Sigma}_{u|\mathcal{L}} + \sigma_y^2\mathbf{I}. \tag{24}$$

2.6 Predictive Distribution

For predictions at new points t^*, the conditional predictive distribution is:

$$u(t^*) \mid \{\mathcal{L}_{\boldsymbol{\lambda}}u = 0, \boldsymbol{\theta}\} \sim \mathcal{N}(\mu_{u|\mathcal{L}}(t^*), \Sigma_{u|\mathcal{L}}(t^*)), \tag{25}$$

$$\mu_{u|\mathcal{L}}(t^*) = \mu_{\boldsymbol{\beta}}(t^*) + \mathrm{Cov}(u^*, \mathcal{L}_{\boldsymbol{\lambda}}u)[\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u, \mathcal{L}_{\boldsymbol{\lambda}}u)]^{-1}(0 - \mathcal{L}_{\boldsymbol{\lambda}}\mu_{\boldsymbol{\beta}}(t)), \tag{26}$$

$$\Sigma_{u|\mathcal{L}}(t^*) = \mathrm{Cov}(u^*,u^*) - \mathrm{Cov}(u^*, \mathcal{L}_{\boldsymbol{\lambda}}u)[\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u, \mathcal{L}_{\boldsymbol{\lambda}}u)]^{-1}\,\mathrm{Cov}(\mathcal{L}_{\boldsymbol{\lambda}}u, u^*). \tag{27}$$

With measurement error incorporated:

$$\Sigma_{y^*|\mathcal{L}}(t^*) = \Sigma_{u|\mathcal{L}}(t^*) + \sigma_y^2\mathbf{I}. \tag{28}$$

This completes the Gaussian process formulation for the exponential growth equation.

3 Illustrative Examples

3.1 Exponential growth

Consider an exponential growth model commonly applied in population dynamics, resource management, and early-stage epidemic modeling (**?**), described as:

$$\frac{\partial u(t)}{\partial t} - \lambda u(t) = 0, \tag{29}$$

with t representing time, $u(t)$ denotes the quantity of interest (population, resource consumption, or contamination level), and $\lambda > 0$ captures the growth rate parameter. The initial condition is $u(0) = 1$. From Equation (29) it follows that $u(t) = \exp\{\lambda t\}$, representing systems where quantities grow exponentially without constraints—a behavior observed in early-stage population growth, bacterial colonies, or initial contamination spread. While simplified, this model serves as a fundamental building block for more complex growth models used in environmental and engineering applications. The exponential growth was simulated using $u(t) = \exp\{\lambda t\} + \epsilon(t)$, with fixed $\lambda = 1$, generating 8 equidistant experimental data points on $t = 0 : 2$, where $\epsilon(t) \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1)$ indicates measurement errors.

To complete the inference setup, the proposed prior distribution is $\lambda \sim \mathcal{HN}(0.5, 1)$ to avoid explosive growth and $\sigma_y \sim \mathcal{HC}(0, 10)$, where $\mathcal{HN}$ and $\mathcal{HC}$ stand for Half-Normal and Half-Cauchy distributions. Figure 2 presents the experimental data alongside the true real growth curve (solid line).

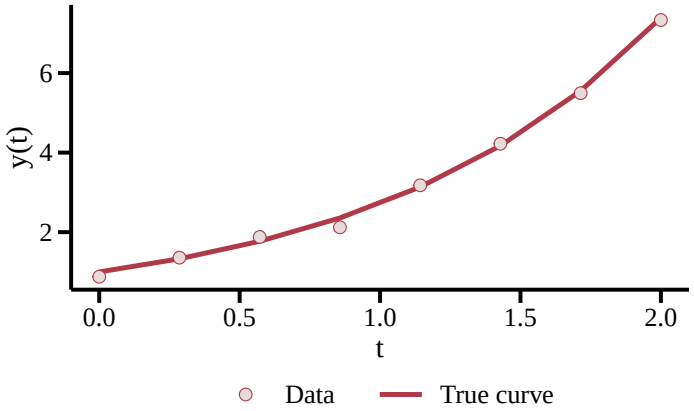

Figure 2: Simulated exponential growth curve. Solid red line: true curve; open white circles: experimental data.

For the exponential growth model, while the analytical solution is readily available, we also implemented a numerical solution using the finite differences method to demonstrate the versatility of our approach. The finite differences approximation discretizes the time domain and approximates the derivatives using difference quotients, resulting in:

$$u(t_{n+1}; \lambda) \approx u(t_n; \lambda) + \Delta t \cdot \lambda\, u(t_n; \lambda),$$

where Δt represents the time step size. This numerical scheme allows us to compare the inference results between exact analytical solutions and numerical approximations, showing that both approaches yield similar parameter estimates for λ. Figure 3 offers a comparative evaluation of nonlinear regression, for both exact and approximated solutions, demonstrating notably similar results.

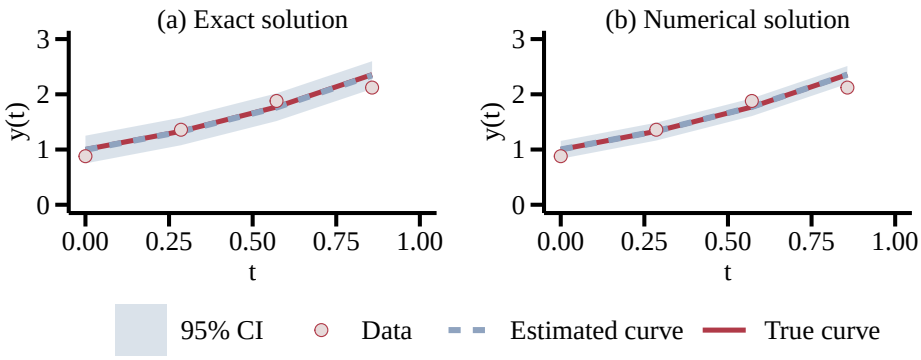

Figure 3: Exponential growth curve estimation via nonlinear regression, zoomed in at $x \in (0, 1)$. (a): exact solution, (b): numerical solution; solid red line: true curve; blue dashed line: estimated curve; shaded blue area: 95% credible interval; open white circles: observed data.

A comparison between the prior and posterior distributions, illustrated in Figure 4, reveals the substantial information gain provided by the observed data in different prior specifications. The Half-Normal prior $\mathcal{HN}(0.5, 1)$ represents moderate skepticism about rapid growth, with a mean of 1.30 and a credible interval 95% of approximately [0.02, 2.0], restricted to the positive domain of the real line. The uniform logarithmic prior, where $\log(\lambda) \sim \text{Unif}(-1, 1)$, spans values of λ from 0.37 to 2.71 on the original scale λ, characterizing exponential decay for the rate, confined within a pre-specified range of possible values. The Half-Cauchy prior $\mathcal{HC}(0, 5)$ with a scale parameter of 5, chosen for its heavy tails and uniform behavior near zero (3), expresses substantial uncertainty with a considerably wider interval.

It is crucial to note that, despite the near identical convergence of the posterior distributions resulting from all priors employed, each prior assumption serves a fundamentally distinct role. Within the framework of PDEs, it is imperative that a prior distribution accurately reflects the physical domain of a specific parameter. This connects directly to Cromwell's rule in Bayesian statistics (**? ?**), which states that one should avoid assigning zero prior probability to any outcome that is not logically impossible. Named after Oliver Cromwell's famous plea for considering the possibility of being mistaken.

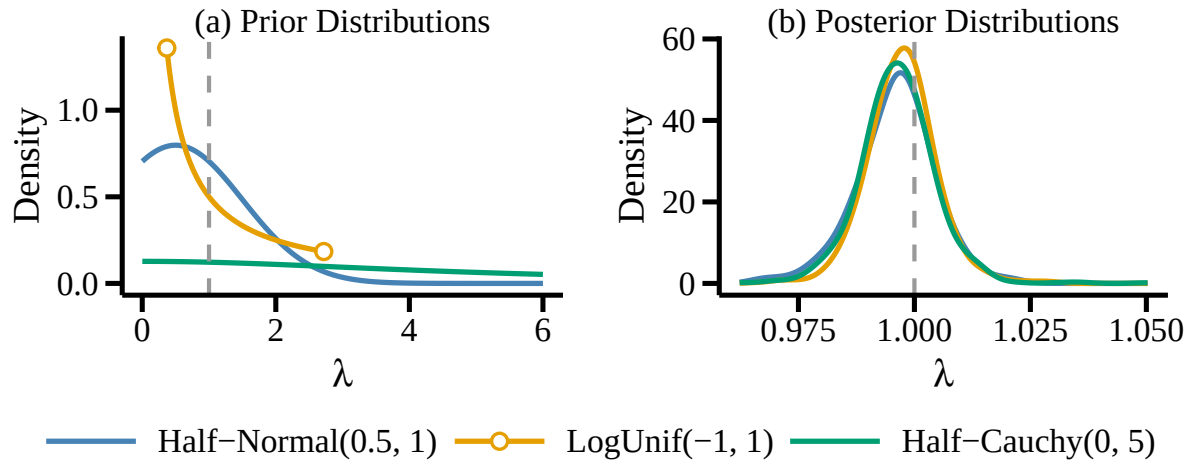

Figure 4: Comparison between (a) prior and (b) posterior distributions for the parameter λ under different prior specifications. Vertical gray dashed line: $\lambda = 1$ true value; Orange open circle: discontinuity point.

3.2 Heat conduction equation

Consider the one-dimensional homogeneous heat conduction equation:

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} - \alpha \frac{\partial^2 u(x,t)}{\partial x^2} &= 0, \\ u(x,0) &= \exp\left(\frac{(x - L/2)^2}{2\sigma_p^2}\right), \\ u(0,t) &= 0, \\ u(L,t) &= 0. \end{aligned} \tag{30}$$

In this context, $t \in \{0, \ldots, T\}$ represents the temporal index, and $x \in \{0, \ldots, L\}$ designates the spatial position within the domain, and α is the thermal diffusivity. The second line of (30) describes a Gaussian pulse centered at the midpoint, with variance σ_p^2. The heat equation finds diverse practical applications (**?**), including diffusion processes in engineering (cooling and heating systems), geophysics (Earth's heat flow), biology (tissue bioheat transfer), finance (option pricing) and materials science (metal processing). Figure 5 provides a simulated visualization of a process governed by the homogeneous heat conduction equation, assuming $\alpha = 1$, $\sigma_p = 1$, $x = 0 : 10$, $t = 0 : 1$, with a dispersion measurement error of 0.1. Heat conduction exhibits a Gaussian distribution, centered at the midpoint of the domain, in accordance with the initial and boundary conditions. For the parameter α, a half-normal prior distribution was assigned, such that $\alpha \sim \mathcal{HN}(0.5, 1)$, which ensures a low prior probability to unrealistic values of α.

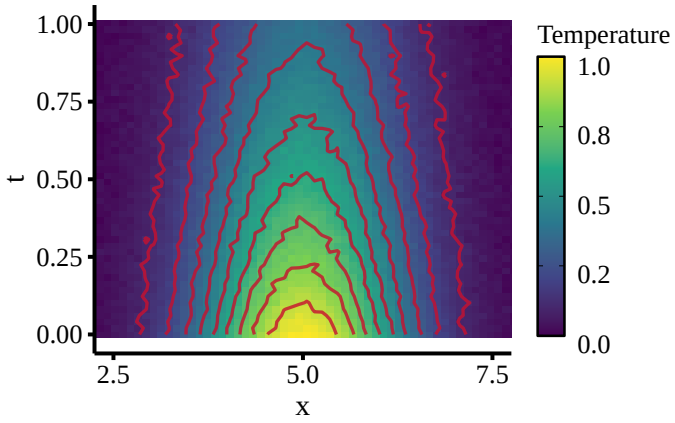

Figure 5: Observed simulated experimental data from the process governed by the heat equation. Thermal diffusivity $\alpha = 1$, $\sigma_p = 1$ and $L = 10$ in Equation (30). Zoomed at the interval (2.5,7.5) in the x-axis.

For the numerical approximation, we employed a second-order finite difference method with the following explicit scheme:

$$u(x, t_{n+1}; \alpha) \approx u(x, t_n; \alpha) + \Delta t \left[\alpha \frac{u(x + \Delta x, t_n) - 2u(x, t_n) + u(x - \Delta x, t_n)}{(\Delta x)^2} \right],$$

where Δt and Δx represent the time and space step sizes, respectively. This numerical scheme must address constraints such as the positivity of the thermal diffusivity parameter α and the Courant-Friedrichs-Lewy (CFL) condition for stability. Our Bayesian framework naturally incorporates these physical and mathematical constraints through the prior distribution on α, which restricts it to physically meaningful values. This prior specification not only enforces physical realism, but also provides regularization that helps mitigate potential numerical instabilities, contributing to the consistency between exact and approximate solution methods observed in our results. The heat conduction equation example demonstrates the framework's ability to handle spatiotemporal processes. The comparison between exact and approximate numerical solutions (Figure 6) reveals the robustness of the Bayesian approach to different computational methods. Both approaches accurately reconstruct the heat diffusion dynamics with minimal differences between them.

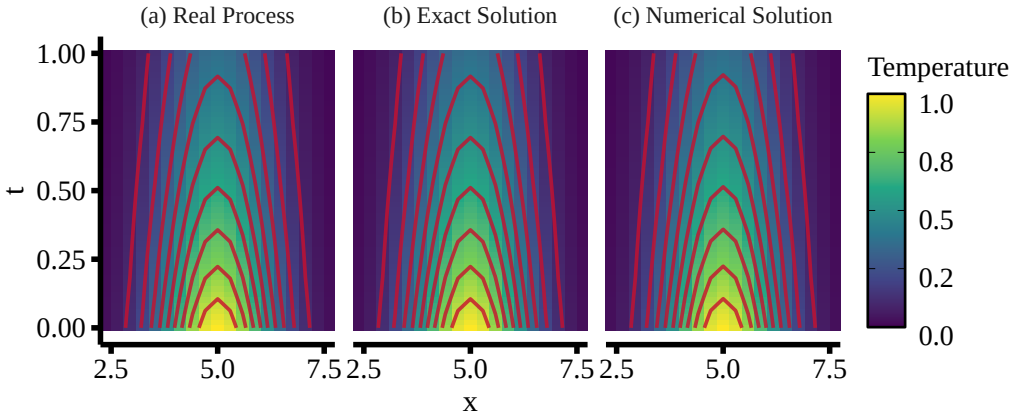

Figure 6: Heat conduction equation solution via nonlinear regression: Real Process vs. Exact and Numerical Solutions; (a): real process; (b): exact solution; (c): numerical solution. Thermal diffusivity $\alpha = 1$, $\sigma_p = 1$ and $L = 10$ in Equation (30). Zoomed at $x \in [2.5, 7.5]$.

The posterior distribution of the diffusion parameter α (Figure 7) associated with the exact solution approach exhibits a narrow concentration around the true value 1.0, with a posterior mean 0.9997 and standard deviation 0.0015, showcasing the effectiveness of the method in addressing the inverse problem of parameter identification. The effective posterior distributions of α remain nearly identical between the exact method and the approximated solution, demonstrating that the inference procedure is not sensitive to the specific solution approach used in the forward model.

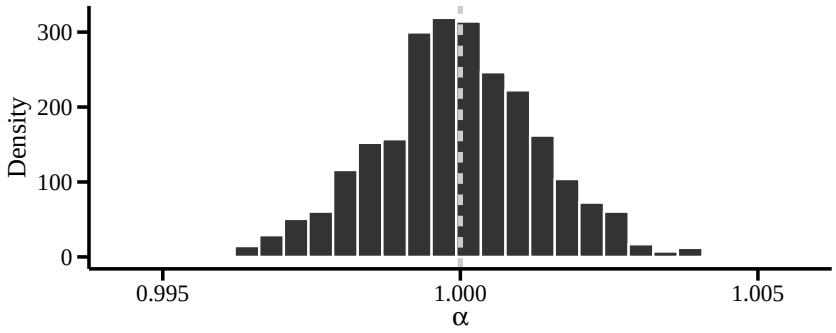

Figure 7: Histogram of the posterior distribution of α; the gray vertical dashed line represents the true value of α.

References

[1] R. M. Neal, MCMC using Hamiltonian dynamics, Handb. Markov Chain Monte Carlo 2 (11) (2011) 113–162.

[2] A. Kucukelbir, R. Ranganath, A. Gelman, D. Blei, Automatic variational inference in Stan, Adv. Neural Inf. Process. Syst. 28 (2015).

[3] A. Gelman, Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper), Bayesian Anal. 1 (3) (2006) 515–534.