DOI: 10.34749/3061-1466.2026.48

Equayes – Democratizing Probabilistic Model Construction and Exploration with automatic Equation to Bayesian Model transformation

Christian Findenig
Embedded Computing and Machine Learning
Materials Center Leoben Forschung GmbH
8700 Leoben, Austria
christian.findenig@mcl.at

Manfred Mücke
Embedded Computing and Machine Learning
Materials Center Leoben Forschung GmbH
8700 Leoben, Austria
manfred.muecke@mcl.at

Abstract

For many scientific and engineering problems, equations based on applicable laws of physics can be used to link observable physical quantities. Analytic expressions, however, provide only point estimates and therefore cannot express uncertainty. This limits trustworthiness of predictions, especially in setups with limited data, noisy observations or when extrapolating. Bayesian probabilistic models address this limitation by treating unknown model parameters as random variables initialized by prior distributions and yielding – through inference – posterior (predictive) distributions. Constructing Bayesian models and convergence of inference, however, still requires specialized knowledge in probabilistic programming and inference algorithms, hindering the broader adoption of Bayesian models and uncertainty quantification in many domains. To make uncertainty-aware equation modeling more accessible, we present **Equayes** (*Equ*ation to B*ayes*ian Model), a scikit-learn-style estimator that converts a user-provided symbolic expression into a probabilistic model and performs posterior inference over its numerical constants. The core value of the method and tool to construction of hybrid models is that it implements a principled approach to hybrid model evaluation, linking laws of physics, random variables and inference in an accessible manner.

1 Introduction

Physics-informed machine learning and hybrid modeling combine mechanistic structure with statistical learning to improve generalization, sample efficiency, and interpretability. In many such workflows, a domain expert already has an analytical equation, a constitutive relation, or a symbolic surrogate, but still needs uncertainty estimates over parameters and predictions for model interpretation and model criticism. This need becomes especially acute when data is scarce or noisy, or when downstream decisions depend on confidence rather than on point estimates alone.

Bayesian modeling provides a principled way to address such scenarios by treating symbols as random variables (rather than point estimates) and combining prior assumptions with observed data to obtain posterior distributions [1]. Since the exact computation of posterior distributions is in general analytically intractable, practitioners resort to approximate or sampling-based inference. Markov chain Monte Carlo (MCMC) methods provide a general framework for drawing samples from the posterior distribution, thereby enabling both the quantification of parameter uncertainty and its propagation to downstream predictions. Recent advances in Hamiltonian Monte Carlo, particularly the No-U-Turn Sampler (NUTS), have substantially improved the robustness of posterior computation in practice [2]. Posterior predictive distributions then combine parameter uncertainty and observation

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

This work is licensed under CC BY 4.0, excluding materials attributed to other sources or explicitly excluded.

noise, which is particularly valuable for uncertainty-aware predictions, model criticism, and error propagation in scientific applications [1].

In practice, however, applying Bayesian methods to some analytic equation still requires substantial manual work. A user must implement the probabilistic program, define latent variables and priors, connect them correctly to the deterministic model, and configure inference machinery in a probabilistic programming framework. Although modern frameworks such as Pyro make Bayesian modeling flexible and expressive [3], they still assume familiarity with probabilistic programming abstractions. Equayes addresses this implementation bottleneck: if a model is already available as a symbolic mathematical expression, then most of the effort required to turn it into a Bayesian model is automated by Equayes.

2 From Equation to Bayesian Model

Let a user-provided symbolic model be written as

$$y = f_\theta(x_1, \ldots, x_N); \quad y, x_i \in \mathbb{R} \tag{1}$$

where $x_1, \ldots, x_N$ denotes the N input variables and $\theta \in \mathbb{R}^K$ denotes numerical constants in the equation. Let further $\mathcal{D} = \{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^M$ be the observations of the system. The goal is to automatically define a probabilistic model $p(y \mid \theta, \mathbf{x})$ and infer the posterior distribution $p(\theta \mid \mathcal{D})$.

Equayes takes the symbolic expression f and

- identifies its numerical constants θ,
- replaces them by latent parameters, i.e. probability distributions $p(\theta)$,
- introduces an observation-noise variable and likelihood $p(y \mid \theta, \mathbf{x})$,
- compiles the resulting expression to a differentiable backend, and then
- runs MCMC inference to sample the posterior $p(y \mid \theta, \mathbf{x})$.

The resulting Bayesian model can be written as

$$\theta_k \sim \mathcal{N}(0, \sigma^2 = 1000), \quad k = 1, \ldots, K \tag{2}$$

$$\tilde{\sigma}^2 \sim \text{HalfNormal}(\sigma^2 = 10) \tag{3}$$

$$y \sim \mathcal{N}(f_\theta(x_1, \ldots, x_N), \tilde{\sigma}^2). \tag{4}$$

The wide variance in the prior over θ (2) encodes that parameters are in general unknown before observing any data. Furthermore, Equayes assumes that the observations are independent and identically distributed, therefore, they are modelled as samples of a Gaussian distribution (4).

In the current implementation, Equayes can be used by only specifying the symbolic expression f, all other parameters are readily set. Using the NUTS inference routine enables Equayes to infer complex posterior distributions over θ without making any assumptions about the geometry of the posterior. Internally, symbolic manipulation is handled through SymPy [4], while probabilistic execution and inference are handled through Pyro [3]. For all steps, Equayes is parameterizable, with default values as follows: `inference_method`: "mcmc", `mcmc_kernel`: "nuts", `mcmc_samples`: 2000, `mcmc_warmup_samples`: 2000, `mcmc_chains`: 1, `mcmc_initial_step_size`: $1e^{-2}$, the starting point of the MCMC chain is set to θ as given in the expression. In general, users of Equayes do not need to adjust those parameters, as they provide a good trade-off between computational cost and quality of the posterior distribution. However, advanced users may customize inference as required.

Executing inference, posterior predictive samples for new inputs, and retrieving posterior samples for θ are then exposed through a scikit-learn-style estimator interface via `fit()`, `predict()`, and `get_posterior()`, respectively. Equayes is available at https://github.com/mclprobability/Equayes.

This design is intentionally focused. Equayes is not intended to replace expert Bayesian modeling in settings that require bespoke priors, hierarchical structure, or custom likelihoods. Even though the architecture of Equayes is extendable to feature more customized models, like customizable prior distributions, which may be available in the future. Instead, it provides a practical and accessible route to uncertainty-aware equation fitting for users who want to preserve an analytical form while

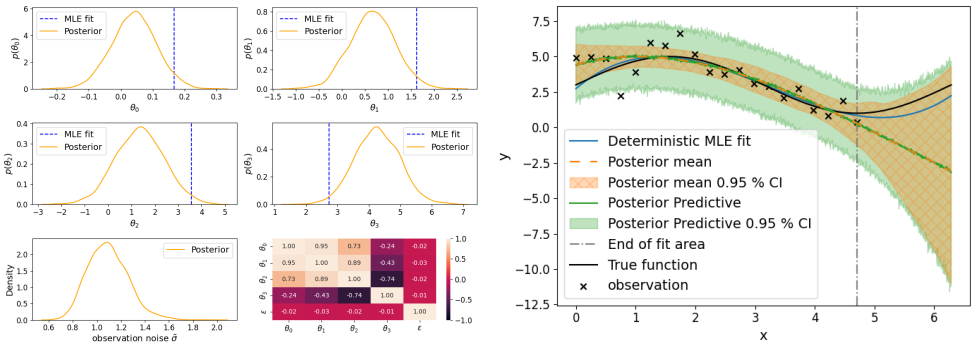

Figure 1: Comparison of deterministic and probabilistic polynomial fit (degree 3). Left: deterministic maximum likelihood estimates (MLE), and marginal posterior distributions and correlation matrix of the inferred parameters. Right: Deterministic fit and Equayes posterior predictive with 95 % credible intervals. The widening credible intervals indicate increasing uncertainty due to limited data support.

obtaining posterior and posterior predictive uncertainty with minimal implementation effort. We believe that this is important across a broad range of scientific and engineering domains, where symbolic relations are often known, inferred, or constrained by prior physical structure, yet the effort required to translate such equations into probabilistic models remains a substantial barrier to applying Bayesian inference in practice [5, 6].

3 Evaluation

First, we demonstrate Equayes on an illustrative example, second, we show the application of Equayes on a materials science use-case (cantilever).

3.1 Illustrative Example

To illustrate the practical gain, we consider noisy observations generated from a sinusoidal ground-truth process and fit a polynomial surrogate of degree 3, $f(x) = \theta_0 x^3 + \theta_1 x^2 + \theta_2 x + \theta_3$. This setup is intentionally misspecified: the polynomial surrogate model cannot represent the true sinusoid exactly, which mirrors most real-world problems in which the available analytical model is useful but not exact. We compare a deterministic maximum-likelihood fit against the posterior predictive result produced by Equayes.

Figure 1 shows that the deterministic solution provides a single reconstruction and therefore hides epistemic uncertainty caused by limited support in parts of the input domain. This uncertainty is visible in the posterior variance (left). Furthermore, the correlation matrix shows the dependence of the parameters under the posterior distribution. The Bayesian fit (right) yields a posterior predictive mean (variance caused by parameter uncertainty only), and full posterior predictive distribution together with 95 % credible intervals. In the example, the credible intervals widen substantially outside of the training region reflecting the uncertainty inherent to extrapolation.

3.2 Cantilever

Consider the Cantilever [7] as representative example of a physical system. Shown in Figure 2 (left) is a prototypical cantilever - a lever fixed on one side. We want to estimate the density ρ of the cantilever, without any possibility of measuring ρ directly. Hence, we need to resort to indirect measurements. The governing equations of the cantilever

$$A = W * H; \quad I = \frac{W * H^3}{12} \tag{5}$$

$$f = \frac{\beta^2}{2\pi L^2}\sqrt{\frac{EI}{\rho A}} \tag{6}$$

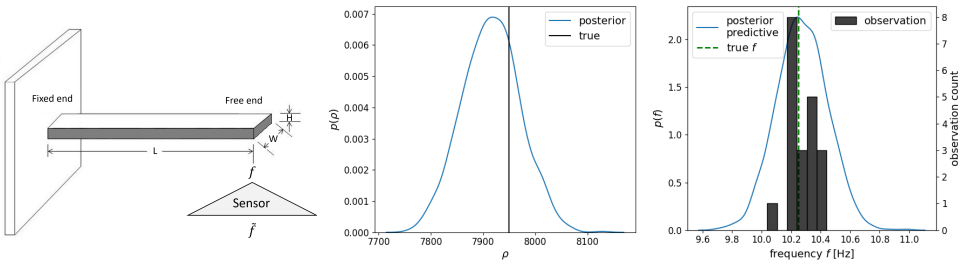

Figure 2: The Cantilever experiment. Left: visualization of the cantilever and measurement setup. Center: the inferred posterior distribution. Right: the posterior predictive distribution (left axis) and histogram of frequency observations (right axis).

with mode dependent β, density ρ, and Young's modulus E establish that the lever's natural frequency f directly depends on its density. We therefore use frequency measurements $\tilde{f}$ as system observations. To model sensor noise, we assume $\tilde{f}$ is distributed according to a Normal distribution, centered at the true frequency f with isotropic Gaussian sensor noise ϵ, $\tilde{f} \sim \mathcal{N}(f, \epsilon^2)$.

We use the governing equation (6) as a generative model to generate a data set of 20 frequency observations with parameters: $L = 0.9$ m, $W = 0.01$ m, $H = 0.01$ m, $\rho = 7950$ kg/m³, $E = 210e^9$ Pa, $\beta = 1.875$, $\epsilon = 0.1$ Hz. Given the generated data, we utilize Equayes to infer the density posterior distribution $p(\rho)$. We implement (6) in sympy and substitute ρ with the assumed density $\rho = 7900$. This tells Equayes to replace the parameter $\rho = 7900$ with a latent variable and expect all other parameters as input. On inference, we leverage the default parameters of Equayes and set all free parameters to their true values.

Figure 2 (center) shows the posterior distribution of density ρ. Its mode almost matches the true density, with widening intervals. This wider posterior distribution $p(\rho)$ expresses the remaining uncertainty given the 20, noisy frequency observations. The posterior predictive distribution (Figure 2 (right)) confirms that the posterior distribution truly describes the observed data, as the predictive distribution closely models the observed data.

4 Conclusion

We introduced Equayes, a scikit-learn-style estimator that turns symbolic equations into Bayesian models with minimal user effort. By automating parameter lifting, noise modeling, symbolic compilation, and MCMC-based inference, Equayes lowers the barrier to probabilistic modeling for equation-centric workflows. The result is not merely a point estimate, but posterior and posterior predictive information that quantify uncertainty and enable more robust downstream use. In this sense, Equayes fills an important gap in the modeling toolbox by making interpretable, prior-aware, and uncertainty-conscious Bayesian treatment of symbolic equations readily accessible in practice.

5 Acknowledgments

Equayes was inspired by a prototype implemented by Bernd Schuscha in Bean Machine to rank sets of equations from symbolic regression. We are most thankful for his work and for continued discussions.

The authors gratefully acknowledge the financial support under the scope of the COMET program within the K2 Center "Integrated Computational Material, Process and Product Engineering (IC-MPPE)" (Project No 886385). This program is supported by the Austrian Federal Ministries for Economy, Energy and Tourism (BMWET) and for Innovation, Mobility and Infrastructure (BMIMI), represented by the Austrian Research Promotion Agency (FFG), and the federal states of Styria, Upper Austria and Tyrol.

ChatGPT 5.4 Thinking was used to reformulate or restructure paragraphs in this extended abstract.

References

[1] Paul Hewson. "Bayesian Data Analysis 3rd edn A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari and D. B. Rubin, 2013 Boca Raton, Chapman and Hall–CRC 676 pp., £44.99 ISBN 1-439-84095-4". In: *Journal of The Royal Statistical Society Series A-statistics in Society* 178 (2015), pp. 301–301. URL: https://api.semanticscholar.org/CorpusID:120872375.

[2] Matthew D. Hoffman and Andrew Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15.47 (2014), pp. 1593–1623. URL: http://jmlr.org/papers/v15/hoffman14a.html.

[3] Eli Bingham et al. "Pyro: Deep Universal Probabilistic Programming". In: *CoRR* abs/1810.09538 (2018). arXiv: 1810.09538. URL: http://arxiv.org/abs/1810.09538.

[4] Aaron Meurer et al. "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3 (Jan. 2017), e103. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.103. URL: https://doi.org/10.7717/peerj-cs.103.

[5] Christopher Krapu and Mark Borsuk. "Probabilistic programming: A review for environmental modellers". In: *Environmental Modelling & Software* 114 (2019), pp. 40–48. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2019.01.014. URL: https://www.sciencedirect.com/science/article/pii/S1364815218308843.

[6] Sai Hung Cheung et al. "Bayesian uncertainty analysis with applications to turbulence modeling". In: *Reliability Engineering & System Safety* 96.9 (2011). Quantification of Margins and Uncertainties, pp. 1137–1149. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2010.09.013. URL: https://www.sciencedirect.com/science/article/pii/S0951832011000664.

[7] N.A. Rubayi and S. Charoenree. "Natural frequencies of vibration of cantilever sandwich beams". In: *Computers & Structures* 7.6 (1977), pp. 737–745. ISSN: 0045-7949. DOI: https://doi.org/10.1016/0045-7949(77)90028-1. URL: https://www.sciencedirect.com/science/article/pii/0045794977900281.