Bayesian Calibration & Inverse Prediction

Mathematical Derivation and Assumptions

McClelland Lab, University College London

February 24, 2026

Contents

1 Introduction

In analytical chemistry and bioassay work a common workflow is to prepare standards at known concentrations, measure the instrument response for each, and fit a calibration curve. The practical goal, however, is the *reverse*: given a new instrument reading, estimate the unknown concentration that produced it.

This is the **inverse prediction** problem. Classical approaches (Fieller's theorem, Wald intervals) provide approximate confidence intervals but rely on asymptotic normality, struggle with nonlinear models, and do not fully propagate parameter uncertainty. A Bayesian treatment resolves these limitations naturally: we obtain the full joint posterior of the model parameters, then push every source of uncertainty—parameter estimation *and* measurement noise—through the inverse to get a complete distribution over the unknown concentration.

This document derives the method implemented in the accompanying Streamlit application.

2 Notation

Symbol	Meaning
n	Number of calibration points
x_i	Known standard value (e.g. concentration) for point i
y_i	Measured instrument response for point i
$\boldsymbol{x}^{\text{cal}} = (x_1, \ldots, x_n)^\top$	Calibration x-values
$\boldsymbol{y}^{\text{cal}} = (y_1, \ldots, y_n)^\top$	Calibration y-values
$f(x;\, \boldsymbol{\theta})$	User-specified calibration function
$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^\top$	Model parameters
σ_y	Baseline noise standard deviation
α	Heteroscedasticity exponent ($\alpha = 0$: constant variance)
A	Scaling constant (geometric mean of calibration y-values)
y^*	New observed response
x^*	Unknown value to be estimated

3 The Calibration Model

We begin with three assumptions that define the generative model for the calibration data.

Assumption 1 (Calibration function)**.** The relationship between the known value x and the instrument response y is described by a parametric function $f : \mathbb{R} \times \mathbb{R}^p \to \mathbb{R}$, specified by the user. For example, $f(x;\, a, b) = a + bx$, or $f(x;\, a, b) = a\, e^{bx}$. We require f to be continuous and differentiable almost everywhere with respect to both x and $\boldsymbol{\theta}$.

Assumption 2 (Gaussian noise)**.** Each observation is the true calibration value plus independent Gaussian noise. In the simplest (homoscedastic) case the noise has constant variance:

$$y_i \;=\; f(x_i;\, \boldsymbol{\theta}) \;+\; \varepsilon_i, \qquad \varepsilon_i \;\sim\; \mathcal{N}(0,\, \sigma_y^2), \qquad i = 1, \ldots, n. \tag{1}$$

The noise terms are mutually independent and independent of the standard values x_i. An extension to non-constant variance is described in Section 4.

Assumption 3 (Differentiability)**.** The log-posterior is differentiable with respect to all continuous parameters. This is guaranteed by the SymPy-parsed equation and PyTensor's automatic differentiation, and is required by the gradient-based NUTS sampler (Section 7).

Under these assumptions the **likelihood** is

$$p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}}, \boldsymbol{\theta}, \sigma_y) \;=\; \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma_y} \exp\!\left[-\frac{\big(y_i - f(x_i;\,\boldsymbol{\theta})\big)^2}{2\,\sigma_y^2}\right], \tag{2}$$

or equivalently, writing $\boldsymbol{\mu} = \big(f(x_1;\boldsymbol{\theta}), \ldots, f(x_n;\boldsymbol{\theta})\big)^\top$,

$$\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{\theta}, \sigma_y \;\sim\; \mathcal{N}\big(\boldsymbol{\mu},\; \sigma_y^2\,\mathbf{I}_n\big). \tag{3}$$

4 Heteroscedastic Variance Model

In many assays—particularly immunoassays and serial dilution experiments—the measurement variance is not constant but increases with the signal level. Following Gelman et al. [2004], the application offers an optional heteroscedastic variance model that replaces the constant-variance assumption (Assumption 2) with a power-law relationship between the variance and the predicted mean.

4.1 Variance Model

Let $\mu_i = f(x_i;\,\boldsymbol{\theta})$ denote the predicted mean response at observation i. The heteroscedastic model is

$$\boxed{y_i \;\sim\; \mathcal{N}\!\left[\mu_i,\;\; \left(\frac{\mu_i}{A}\right)^{2\alpha} \sigma_y^2\right], \qquad i = 1, \ldots, n,} \tag{4}$$

where:

- A is a scaling constant set to the geometric mean of the observed calibration responses, $A = \exp\!\big(\frac{1}{n}\sum_{i=1}^{n} \log y_i\big)$. Its role is to make σ_y interpretable as the noise standard deviation at a "typical" measurement level.
- $\alpha \geq 0$ controls how variance scales with the mean:
  - $\alpha = 0$: constant variance (reduces to the homoscedastic model).
  - $\alpha = 1$: variance proportional to μ_i^2, i.e. approximately constant coefficient of variation (CV).
  - $\alpha \in (0, 2)$: intermediate and super-proportional variance structures.

The observation-specific standard deviation is therefore

$$\sigma_i \;=\; \left|\frac{\mu_i}{A}\right|^{\alpha} \sigma_y\,, \tag{5}$$

and the heteroscedastic likelihood becomes

$$p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}}, \boldsymbol{\theta}, \sigma_y, \alpha) \;=\; \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma_i} \exp\!\left[-\frac{(y_i - \mu_i)^2}{2\,\sigma_i^2}\right]. \tag{6}$$

4.2 Prior on α

We assign α a uniform prior on $[0, 2]$:

$$\alpha \;\sim\; \text{Uniform}(0,\, 2). \tag{7}$$

This range is centred at proportionality ($\alpha = 1$) and is wide enough to accommodate the constant-variance special case ($\alpha \approx 0$) as well as super-proportional variance. The data determine the posterior for α; when constant variance is adequate, the posterior will concentrate near zero.

5 Prior Distributions

The application allows the user to choose prior distributions for all parameters through the Advanced Options panel. The defaults are weakly informative, independent priors:

$$\theta_j \sim \mathcal{N}(0,\, 10^2), \qquad j = 1, \ldots, p, \tag{8}$$

$$\sigma_y \sim \mathcal{N}^+(10), \tag{9}$$

where $\mathcal{N}^+(\tau)$ denotes the half-normal distribution with scale τ, i.e. a $\mathcal{N}(0, \tau^2)$ truncated to the positive reals.

Supported prior families. For each parameter the user may select from: Normal, Half-Normal, Uniform, or Log-Normal distributions, each with user-specified hyperparameters.

5.1 Log-Scale Parameterisation

For parameters that must be positive (e.g. rate constants, asymptotes), the user may opt to model the parameter on the log scale. For a parameter $\theta_j > 0$ this means:

$$\log \theta_j \;\sim\; \pi(\cdot), \qquad \theta_j \;=\; \exp(\log \theta_j), \tag{10}$$

where $\pi(\cdot)$ is the chosen prior placed on the unconstrained $\log \theta_j$. This enforces positivity without requiring bounded priors and improves sampling geometry for parameters that span several orders of magnitude.

Rationale. The zero-centred normal priors on each θ_j are deliberately vague (standard deviation 10). The half-normal prior on σ_y enforces positivity while remaining uninformative over the plausible range of noise magnitudes. These defaults perform well across a broad class of calibration problems. Users with strong domain knowledge should substitute tighter priors via the Advanced Options panel.

Joint prior. Because the priors are independent,

$$p(\boldsymbol{\theta},\, \sigma_y,\, \alpha) \;=\; \left[\prod_{j=1}^{p} p(\theta_j)\right] p(\sigma_y)\; p(\alpha). \tag{11}$$

(When the homoscedastic model is used, the $p(\alpha)$ factor is absent and α is not part of the model.)

6 Posterior Distribution

Applying Bayes' theorem:

$$\boxed{p(\boldsymbol{\theta},\, \sigma_y,\, \alpha \mid \boldsymbol{y}^{\text{cal}},\, \boldsymbol{x}^{\text{cal}}) \;=\; \frac{p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}},\, \boldsymbol{\theta},\, \sigma_y,\, \alpha)\;\; p(\boldsymbol{\theta},\, \sigma_y,\, \alpha)}{p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}})}} \tag{12}$$

where the marginal likelihood (evidence) is

$$p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}}) \;=\; \int p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}},\, \boldsymbol{\theta},\, \sigma_y,\, \alpha)\; p(\boldsymbol{\theta},\, \sigma_y,\, \alpha)\; \mathrm{d}\boldsymbol{\theta}\; \mathrm{d}\sigma_y\; \mathrm{d}\alpha. \tag{13}$$

For most nonlinear calibration functions this integral is analytically intractable, so we approximate the posterior using Markov chain Monte Carlo sampling. (When the homoscedastic model is used, α is absent from all expressions.)

7 MCMC Sampling via NUTS

7.1 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) [Neal, 2011] augments the parameter space with auxiliary momentum variables $\boldsymbol{r} \in \mathbb{R}^d$ (where $d = p + 1$) and defines the joint density

$$p(\boldsymbol{\theta}, \sigma, \boldsymbol{r}) \;\propto\; \exp\!\big[-U(\boldsymbol{\theta}, \sigma) - \tfrac{1}{2}\,\boldsymbol{r}^\top \mathbf{M}^{-1} \boldsymbol{r}\,\big], \tag{14}$$

with *potential energy*

$$U(\boldsymbol{\theta}, \sigma) \;=\; -\log p(\boldsymbol{y}^{\text{cal}} \mid \boldsymbol{x}^{\text{cal}}, \boldsymbol{\theta}, \sigma) \;-\; \log p(\boldsymbol{\theta}, \sigma) \tag{15}$$

and mass matrix $\mathbf{M}$ (adapted during warm-up to approximate the posterior covariance).

HMC simulates Hamiltonian dynamics using the leapfrog integrator with step size ϵ:

$$\boldsymbol{r}_{t+\epsilon/2} = \boldsymbol{r}_t - \tfrac{\epsilon}{2}\,\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta}_t), \tag{16}$$

$$\boldsymbol{\theta}_{t+\epsilon} = \boldsymbol{\theta}_t + \epsilon\, \mathbf{M}^{-1}\, \boldsymbol{r}_{t+\epsilon/2}, \tag{17}$$

$$\boldsymbol{r}_{t+\epsilon} = \boldsymbol{r}_{t+\epsilon/2} - \tfrac{\epsilon}{2}\,\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta}_{t+\epsilon}). \tag{18}$$

After L leapfrog steps the proposal is accepted with probability $\min\!\big(1, \exp(-\Delta H)\big)$, where ΔH is the change in the Hamiltonian.

7.2 The No-U-Turn Sampler

NUTS [Hoffman and Gelman, 2014] removes the need to choose L by building a balanced binary tree of leapfrog steps. The tree doubles in size until a "U-turn" is detected:

$$\boldsymbol{r} \cdot (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-) < 0 \qquad \text{or} \qquad \boldsymbol{r} \cdot (\boldsymbol{\theta}^- - \boldsymbol{\theta}^+) < 0, \tag{19}$$

where $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ are the trajectory endpoints. A multinomial scheme selects the next state from the trajectory, weighted by the unnormalised density. The step size ϵ is tuned during warm-up via dual averaging [Nesterov, 2009] to target an acceptance rate of ~ 0.8.

7.3 Warm-up, Sampling, and Diagnostics

1. **Warm-up.** The step size and mass matrix are adapted; these draws are discarded.
2. **Sampling.** S draws are collected from each of C independent chains, giving $N = S \times C$ posterior samples $\{(\boldsymbol{\theta}^{(s)}, \sigma^{(s)})\}_{s=1}^{N}$.
3. **Convergence checks.** The application reports:
   - $\hat{R}$ (Gelman–Rubin statistic): values $\lesssim 1.01$ indicate convergence [Gelman et al., 2013].
   - Effective sample size (n_{eff}): the number of effectively independent draws after accounting for autocorrelation.
   - Monte Carlo standard error (MCSE): the precision of the posterior mean estimate.

8 Inverse Prediction

8.1 Problem Statement

Given a new instrument reading y^*, we want the posterior predictive distribution of the unknown value x^* that produced it. We assume the new observation arises from the same process as the calibration data.

Homoscedastic case.

$$y^* \;=\; f(x^*; \boldsymbol{\theta}) \;+\; \varepsilon^*, \qquad \varepsilon^* \sim \mathcal{N}(0, \sigma_y^2). \tag{20}$$

Heteroscedastic case. The noise standard deviation at the new point depends on the predicted mean $\mu^* = f(x^*; \boldsymbol{\theta})$, which is unknown. We approximate it by using the observed value y^* as a plug-in for μ^*:

$$\varepsilon^* \sim \mathcal{N}\Big(0,\ \big|y^*/A\big|^{2\alpha}\,\sigma_y^2\Big). \tag{21}$$

This approximation is accurate when y^* is close to μ^* (i.e. when the signal-to-noise ratio is moderate to high).

8.2 Posterior Predictive Distribution of x^*

The target distribution is obtained by marginalising over the posterior:

$$\boxed{p(x^* \mid y^*, \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}}) \;=\; \int p(x^* \mid y^*, \boldsymbol{\theta}, \sigma_y, \alpha)\ \ p(\boldsymbol{\theta}, \sigma_y, \alpha \mid \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}})\ \mathrm{d}\boldsymbol{\theta}\ \mathrm{d}\sigma_y\ \mathrm{d}\alpha} \tag{22}$$

This integral propagates *both* parameter uncertainty and measurement noise into the prediction. We evaluate it by Monte Carlo:

Proposition 1 (Monte Carlo inverse prediction)**.** *For each posterior draw* $(\boldsymbol{\theta}^{(s)}, \sigma_y^{(s)}, \alpha^{(s)})$, $s = 1, \ldots, N$*:*

1. ***Compute noise scale:*** $\sigma^{*(s)} = \big|y^*/A\big|^{\alpha^{(s)}}\,\sigma_y^{(s)}$ *(or simply* $\sigma^{*(s)} = \sigma_y^{(s)}$ *in the homoscedastic case).*
2. ***Add noise:*** $\tilde{y}^{(s)} = y^* + \epsilon^{(s)}$ *where* $\epsilon^{(s)} \sim \mathcal{N}(0,\ \sigma^{*(s)2})$.
3. ***Invert:*** $x^{*(s)} = f^{-1}\big(\tilde{y}^{(s)};\ \boldsymbol{\theta}^{(s)}\big)$.

The resulting collection $\{x^{*(s)}\}_{s=1}^{N}$ *is a sample from* $p(x^* \mid y^*, \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}})$.

Proof. For fixed $(\boldsymbol{\theta}, \sigma_y, \alpha)$ the forward model is deterministic, so $x^* = f^{-1}(y^* - \varepsilon^*; \boldsymbol{\theta})$ with $\varepsilon^* \sim \mathcal{N}(0, \sigma^{*2})$. Drawing $\tilde{y}^{(s)} = y^* + \epsilon^{(s)}$ where $\epsilon^{(s)} \sim \mathcal{N}(0, \sigma^{*(s)2})$ is equivalent to sampling the noise-free response from the predictive distribution at draw s. Since the draws $(\boldsymbol{\theta}^{(s)}, \sigma_y^{(s)}, \alpha^{(s)})$ come from the posterior $p(\boldsymbol{\theta}, \sigma_y, \alpha \mid \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}})$, the composition produces samples from the marginal (22). □

8.3 Inversion Methods

Symbolic inverse. When f is algebraically invertible, SymPy computes $x = f^{-1}(y; \boldsymbol{\theta})$ in closed form. If multiple real solutions exist (e.g. for a quadratic), the one closest to the centroid of the calibration x-values is selected.

Numerical inverse. When no closed form exists, Brent's root-finding method [Brent, 1973] solves

$$f(x; \boldsymbol{\theta}^{(s)}) - \tilde{y}^{(s)} = 0 \tag{23}$$

over the interval $\big[x_{\min} - 3\,\Delta x,\ \ x_{\max} + 3\,\Delta x\big]$ where $\Delta x = x_{\max} - x_{\min}$. Brent's method combines bisection, secant, and inverse quadratic interpolation, guaranteeing convergence whenever a sign change exists.

8.4 Credible Intervals and Point Estimates

From the N draws $\{x^{*(s)}\}$ (after discarding any non-finite values from failed inversions), the $100(1-\alpha)\%$ equal-tailed credible interval is

$$\text{CI}_{1-\alpha} \;=\; \big[\,Q_{\alpha/2},\ \ Q_{1-\alpha/2}\,\big], \tag{24}$$

where Q_q is the q-th sample quantile. The application also reports:

$$\hat{x}_{\text{median}} = Q_{0.5}, \tag{25}$$

$$\hat{x}_{\text{mean}} = \frac{1}{N}\sum_{s=1}^{N} x^{*(s)}, \tag{26}$$

$$\hat{\sigma}_x = \sqrt{\frac{1}{N-1}\sum_{s=1}^{N}\left(x^{*(s)} - \hat{x}_{\text{mean}}\right)^2}. \tag{27}$$

9 Residual Diagnostics

After fitting the model, the application runs automated diagnostics to help the user assess whether the chosen equation and the noise assumptions are adequate.

9.1 Residuals

Residuals are evaluated at the posterior median parameters $\hat{\boldsymbol{\theta}} = \text{median}\{\boldsymbol{\theta}^{(s)}\}$:

$$e_i \;=\; y_i \;-\; f(x_i;\,\hat{\boldsymbol{\theta}}), \qquad i = 1, \ldots, n. \tag{28}$$

Two plots are shown: residuals versus fitted values $\hat{y}_i$ and residuals versus x_i.

9.2 Breusch–Pagan Test

The Breusch–Pagan test [Breusch and Pagan, 1979] checks whether the variance of the residuals depends on the fitted values:

$$H_0:\ \text{Var}(\varepsilon_i) = \sigma^2 \ \ \forall\, i \qquad \text{vs.} \qquad H_1:\ \text{Var}(\varepsilon_i) = h(\boldsymbol{z}_i^{\top}\boldsymbol{\gamma}). \tag{29}$$

The procedure regresses the squared residuals e_i^2 on the fitted values by OLS and computes the LM statistic $\text{LM} = \frac{1}{2}\,\text{ESS}$, which follows a χ^2_1 distribution under H_0. A p-value below 0.05 indicates that the noise spread changes with signal level. In that case, the application advises applying a variance-stabilising transformation (e.g. $\log y$ or $\sqrt{y}$) before re-fitting.

9.3 Wald–Wolfowitz Runs Test

The runs test [Wald and Wolfowitz, 1940] assesses whether the sign pattern of the residuals $(+/-)$ is random. Let n_+ and n_- be the counts of positive and negative residuals, and R the number of runs. Under H_0 (random ordering):

$$\mathbb{E}[R] = 1 + \frac{2\,n_+\,n_-}{n_+ + n_-}\,, \tag{30}$$

$$\text{Var}(R) = \frac{2\,n_+\,n_-\,(2\,n_+\,n_- - n_+ - n_-)}{(n_+ + n_-)^2\,(n_+ + n_- - 1)}\,. \tag{31}$$

The standardised statistic $Z = (R - \mathbb{E}[R])/\sqrt{\text{Var}(R)}$ is approximately standard normal. A p-value below 0.05 suggests systematic structure that the model does not capture; the user should consider a different functional form.

9.4 Practical Guidance

Based on the diagnostics the application presents the following advice:

- **Curved / trending residuals:** try a different model (e.g. add a polynomial term, switch to an exponential or power law).
- **Fan-shaped spread:** enable the **heteroscedastic variance model** in Advanced Options, which lets the model learn how noise scales with the mean response. Alternatively, try a variance-stabilising transformation such as $\log(y)$ or $\sqrt{y}$ before fitting.
- **Isolated outliers:** check for data-entry errors; consider excluding the suspect point and re-fitting.
- **Random scatter around zero:** the model assumptions appear satisfied.

10 Summary of Assumptions

1. The user-specified function $f(x;\boldsymbol{\theta})$ adequately describes the calibration relationship (Assumption 1).
2. Observation noise is additive and Gaussian, with either constant variance σ_y^2 or signal-dependent variance $(\mu_i/A)^{2\alpha}\sigma_y^2$ (Assumption 2, Section 4).
3. The log-posterior is differentiable with respect to all continuous parameters (Assumption 3).
4. Prior distributions are weakly informative and mutually independent (user-configurable via Advanced Options; Section 5).
5. New measurements follow the same generative process as the calibration data (no distribution shift).

When assumptions may be violated.

- *Wrong functional form.* The runs test will flag non-random residual patterns; try a different equation.
- *Non-constant variance.* The Breusch–Pagan test will flag this; enable the heteroscedastic variance model or apply a variance-stabilising transform (e.g. $\log y$) and re-fit.
- *Non-Gaussian noise.* Heavy tails may cause credible intervals to undercover. A Student-t likelihood could be added in future.
- *Correlated observations.* If measurements are time-dependent, the independence assumption is violated and intervals will be too narrow.

11 Implementation Notes

- **Equation parsing:** SymPy with implicit multiplication and `convert_xor` transformations.
- **Automatic differentiation:** PyTensor provides exact gradients for the NUTS sampler.
- **Sampling:** PyMC ≥ 5.10 with the default NUTS implementation.
- **Diagnostics:** ArviZ for convergence statistics; statsmodels for the Breusch–Pagan test; the Wald–Wolfowitz test is implemented directly using SciPy.
- **Numerical inversion:** SciPy's `brentq` with tolerance 10^{-10}.

References

T. S. Breusch and A. R. Pagan. A simple test for heteroscedasticity and random coefficient variation. *Econometrica*, 47(5):1287–1294, 1979.

R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, 1973.

A. Gelman, G. L. Chew, and M. Shnaidman. Bayesian analysis of serial dilution assays. *Biometrics*, 60(2):407–417, 2004.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 3rd edition, 2013.

M. D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, chapter 5. Chapman & Hall/CRC, 2011.

Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *The Annals of Mathematical Statistics*, 11(2):147–162, 1940.