

# Bayesian Calibration & Inverse Prediction

Mathematical Derivation and Assumptions

McClelland Lab, University College London

February 25, 2026

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Notation</b>	<b>2</b>
<b>3</b>	<b>The Calibration Model</b>	<b>2</b>
<b>4</b>	<b>Heteroscedastic Variance Model</b>	<b>3</b>
4.1	Built-in Variance Functions . . . . .	3
<b>5</b>	<b>Prior Distributions</b>	<b>4</b>
5.1	Log-Scale Parameterisation . . . . .	4
<b>6</b>	<b>Posterior Distribution</b>	<b>4</b>
<b>7</b>	<b>MCMC Sampling via NUTS</b>	<b>5</b>
7.1	Hamiltonian Monte Carlo . . . . .	5
7.2	The No-U-Turn Sampler . . . . .	5
7.3	Warm-up, Sampling, and Diagnostics . . . . .	5
<b>8</b>	<b>Inverse Prediction</b>	<b>5</b>
8.1	Problem Statement . . . . .	5
8.2	Posterior Predictive Distribution of $x^*$ . . . . .	6
8.3	Inversion Methods . . . . .	6
8.4	Credible Intervals and Point Estimates . . . . .	6
<b>9</b>	<b>Residual Diagnostics</b>	<b>7</b>
9.1	Residuals . . . . .	7
9.2	Breusch–Pagan Test . . . . .	7
9.3	Practical Guidance . . . . .	7
<b>10</b>	<b>Summary of Assumptions</b>	<b>8</b>
<b>11</b>	<b>Implementation Notes</b>	<b>8</b>

---

## 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 obtain a complete distribution over the unknown value. Gelman et al. [2004] demonstrate this approach for immunoassay serial dilution data; the method implemented here generalises it to arbitrary user-specified calibration functions and variance models.

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 $i$
$\mathbf{x}^{\text{cal}} = (x_1, \dots, x_n)^{\top}$	Calibration $x$ -values
$\mathbf{y}^{\text{cal}} = (y_1, \dots, y_n)^{\top}$	Calibration $y$ -values
$f(x; \boldsymbol{\theta})$	User-specified calibration (mean) function
$\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^{\top}$	Mean model parameters
$\sigma_y$	Baseline noise standard deviation
$g(\mu, \sigma_y; \boldsymbol{\phi})$	User-specified noise standard deviation function
$\boldsymbol{\phi}$	Additional variance model parameters (may be empty)
$y^*$	New observed response
$x^*$	Unknown value to be estimated
$\alpha$	Tail probability for credible intervals ( $\alpha = 0.025$ for 95% CI)

## 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  $f : \mathbb{R} \times \mathbb{R}^p \rightarrow \mathbb{R}$ , 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 equals the true mean response plus independent Gaussian noise. In the simplest (homoscedastic) case the noise has constant standard deviation  $\sigma_y$ :

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

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

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

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

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

## 4 Heteroscedastic Variance Model

In many assays the measurement noise is not constant but varies systematically with the signal level. The application allows the user to replace the constant-variance assumption (Assumption 2) with any symbolic variance equation of the form

$$\sigma_i = g(\mu_i, \sigma_y; \boldsymbol{\phi}), \quad (4)$$

where  $\mu_i = f(x_i; \boldsymbol{\theta})$  is the predicted mean at observation  $i$ ,  $\sigma_y > 0$  is a baseline noise scale, and  $\boldsymbol{\phi}$  is a (possibly empty) vector of additional variance parameters. The observation model becomes

$$y_i \sim \mathcal{N}(\mu_i, \sigma_i^2), \quad \sigma_i = g(\mu_i, \sigma_y; \boldsymbol{\phi}), \quad i = 1, \dots, n, \quad (5)$$

and the heteroscedastic likelihood is

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

### 4.1 Built-in Variance Functions

The application provides three preset choices and a free-form custom entry:

Name	$g(\mu_i, \sigma_y; \boldsymbol{\phi})$	Notes
Constant	$\sigma_y$	Homoscedastic; $\boldsymbol{\phi} = \emptyset$
Proportional to mean	$\mu_i \sigma_y$	Constant coefficient of variation (CV)
Power-law	$ \mu_i/A ^\gamma \sigma_y$	$A$ : geometric mean of $\mathbf{y}^{\text{cal}}$ ; $\boldsymbol{\phi} = \{\gamma\}$

In the power-law model,  $\gamma \geq 0$  controls how the noise scales with the mean ( $\gamma = 0$ : constant variance;  $\gamma = 1$ : constant CV;  $\gamma > 1$ : super-proportional variance), and  $A = \exp(\frac{1}{n} \sum_{i=1}^n \log y_i)$  is a fixed scaling constant that makes  $\sigma_y$  interpretable as the noise standard deviation at a typical measurement level. Gelman et al. [2004] used this power-law form to model serial dilution immunoassay data; it subsumes the constant and proportional presets as special cases. For the power-law model we assign  $\gamma$  a uniform prior on  $[0, 2]$ :

$$\gamma \sim \text{Uniform}(0, 2). \quad (7)$$

More generally, priors on  $\boldsymbol{\phi}$  are specified by the user in the same way as priors on  $\boldsymbol{\theta}$  (see Section 5).

## 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:

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

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

where  $\mathcal{N}^+(\tau)$  denotes the half-normal distribution with scale  $\tau$ , 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, Log-Normal, Exponential, or Gamma 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:

$$\log \theta_j \sim \pi(\cdot), \quad \theta_j = \exp(\log \theta_j), \quad (10)$$

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  $\theta_j$  are deliberately vague (standard deviation 10). The half-normal prior on  $\sigma_y$  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,

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

When the constant-variance model is used  $\boldsymbol{\phi}$  is empty and the last factor is absent.

## 6 Posterior Distribution

Applying Bayes' theorem:

$$p(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi} | \mathbf{y}^{\text{cal}}, \mathbf{x}^{\text{cal}}) = \frac{p(\mathbf{y}^{\text{cal}} | \mathbf{x}^{\text{cal}}, \boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi}) p(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi})}{p(\mathbf{y}^{\text{cal}} | \mathbf{x}^{\text{cal}})} \quad (12)$$

where the marginal likelihood (evidence) is

$$p(\mathbf{y}^{\text{cal}} | \mathbf{x}^{\text{cal}}) = \int p(\mathbf{y}^{\text{cal}} | \mathbf{x}^{\text{cal}}, \boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi}) p(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi}) d\boldsymbol{\theta} d\sigma_y d\boldsymbol{\phi}. \quad (13)$$

For most nonlinear calibration functions this integral is analytically intractable, so we approximate the posterior using Markov chain Monte Carlo sampling.

## 7 MCMC Sampling via NUTS

### 7.1 Hamiltonian Monte Carlo

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

$$p(\boldsymbol{\theta}, \sigma_y, \mathbf{r}) \propto \exp[-U(\boldsymbol{\theta}, \sigma_y) - \frac{1}{2} \mathbf{r}^\top \mathbf{M}^{-1} \mathbf{r}], \quad (14)$$

with *potential energy*

$$U(\boldsymbol{\theta}, \sigma_y) = -\log p(\mathbf{y}^{\text{cal}} | \mathbf{x}^{\text{cal}}, \boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi}) - \log p(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi}) \quad (15)$$

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  $\epsilon$ :

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

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

$$\mathbf{r}_{t+\epsilon} = \mathbf{r}_{t+\epsilon/2} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}_{t+\epsilon}). \quad (18)$$

After  $L$  leapfrog steps the proposal is accepted with probability  $\min(1, \exp(-\Delta H))$ , where  $\Delta H$  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:

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

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  $\epsilon$  is tuned during warm-up via dual averaging [Nesterov, 2009] to target an acceptance rate of  $\sim 0.8$ .

### 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  $\{(\boldsymbol{\theta}^{(s)}, \sigma_y^{(s)}, \boldsymbol{\phi}^{(s)})\}_{s=1}^N$ .
3. **Convergence checks.** The application reports:
  - $\hat{R}$  (Gelman–Rubin statistic): values  $\lesssim 1.01$  indicate convergence [Gelman et al., 2013].
  - Effective sample size ( $n_{\text{eff}}$ ): 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:

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

where the noise standard deviation is

$$\sigma^* = g(\mu^*, \sigma_y; \phi), \quad \mu^* = f(x^*; \theta). \quad (21)$$

Because  $x^*$  is unknown,  $\mu^*$  is also unknown. We approximate it by substituting the observed value  $y^*$  as a plug-in for  $\mu^*$ :

$$\sigma^* \approx g(y^*, \sigma_y; \phi). \quad (22)$$

This approximation is accurate when the signal-to-noise ratio is moderate to high (i.e.  $y^* \approx \mu^*$ ). In the constant-variance case  $g = \sigma_y$  and no approximation is needed.

## 8.2 Posterior Predictive Distribution of $x^*$

The target distribution is obtained by marginalising over the posterior:

$$p(x^* | y^*, \mathbf{y}^{\text{cal}}, \mathbf{x}^{\text{cal}}) = \int p(x^* | y^*, \boldsymbol{\theta}, \sigma_y, \phi) p(\boldsymbol{\theta}, \sigma_y, \phi | \mathbf{y}^{\text{cal}}, \mathbf{x}^{\text{cal}}) d\boldsymbol{\theta} d\sigma_y d\phi \quad (23)$$

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  $(\boldsymbol{\theta}^{(s)}, \sigma_y^{(s)}, \phi^{(s)})$ ,  $s = 1, \dots, N$ :*

1. **Compute noise scale:**  $\sigma^{*(s)} = g(y^*, \sigma_y^{(s)}; \phi^{(s)})$  (using the plug-in approximation (22)).
2. **Add noise:**  $\tilde{y}^{(s)} = y^* + \epsilon^{(s)}$  where  $\epsilon^{(s)} \sim \mathcal{N}(0, \sigma^{*(s)2})$ .
3. **Invert:**  $x^{*(s)} = f^{-1}(\tilde{y}^{(s)}; \boldsymbol{\theta}^{(s)})$ .

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

*Proof.* Fix a posterior draw  $(\boldsymbol{\theta}, \sigma_y, \phi)$ . By the observation model (20), the unknown satisfies  $x^* = f^{-1}(y^* - \varepsilon^*; \boldsymbol{\theta})$  where  $\varepsilon^* \sim \mathcal{N}(0, \sigma^{*2})$ . Because the Gaussian distribution is symmetric,  $-\varepsilon^* \stackrel{d}{=} \varepsilon^*$ , so drawing  $\tilde{y}^{(s)} = y^* + \epsilon^{(s)}$  with  $\epsilon^{(s)} \sim \mathcal{N}(0, \sigma^{*(s)2})$  and setting  $x^{*(s)} = f^{-1}(\tilde{y}^{(s)}; \boldsymbol{\theta}^{(s)})$  is equivalent in distribution to sampling  $x^* = f^{-1}(y^* - \varepsilon^*; \boldsymbol{\theta})$ . Since the draws come from the posterior  $p(\boldsymbol{\theta}, \sigma_y, \phi | \mathbf{y}^{\text{cal}}, \mathbf{x}^{\text{cal}})$ , the composition produces samples from the marginal (23).  $\square$

## 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

$$f(x; \boldsymbol{\theta}^{(s)}) - \tilde{y}^{(s)} = 0 \quad (24)$$

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

## 8.4 Credible Intervals and Point Estimates

From the  $N$  draws  $\{x^{*(s)}\}$  (after discarding any non-finite values from failed inversions), the  $100(1 - 2\alpha)\%$  equal-tailed credible interval is

$$\text{CI}_{1-2\alpha} = [Q_\alpha, Q_{1-\alpha}], \quad (25)$$

where  $Q_q$  denotes the  $q$ -th sample quantile and  $\alpha$  is the tail probability (e.g.  $\alpha = 0.025$  for a 95% interval). The application also reports:

$$\hat{x}_{\text{median}} = Q_{0.5}, \quad (26)$$

$$\hat{x}_{\text{mean}} = \frac{1}{N} \sum_{s=1}^N x^{*(s)}, \quad (27)$$

$$\hat{\sigma}_x = \sqrt{\frac{1}{N-1} \sum_{s=1}^N (x^{*(s)} - \hat{x}_{\text{mean}})^2}. \quad (28)$$

## 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  $\hat{\boldsymbol{\theta}} = \text{median}\{\boldsymbol{\theta}^{(s)}\}$  and posterior median variance parameters  $\hat{\boldsymbol{\phi}}$ :

$$e_i = y_i - f(x_i; \hat{\boldsymbol{\theta}}), \quad i = 1, \dots, n. \quad (29)$$

Standardised residuals are obtained by dividing by the model-predicted noise standard deviation at each point:

$$r_i = \frac{e_i}{\hat{\sigma}_i}, \quad \hat{\sigma}_i = g(\hat{\mu}_i, \hat{\sigma}_y; \hat{\boldsymbol{\phi}}), \quad \hat{\mu}_i = f(x_i; \hat{\boldsymbol{\theta}}). \quad (30)$$

Two plots are shown: raw residuals  $e_i$  versus  $x_i$  (with posterior prediction intervals) and standardised residuals  $r_i$  versus  $x_i$ . If the variance model is adequate, the standardised residuals should scatter randomly around zero with roughly constant spread.

### 9.2 Breusch–Pagan Test

The Breusch–Pagan test [Breusch and Pagan, 1979] checks whether the *standardised* residuals  $r_i$  still exhibit variance that depends on  $x_i$ , i.e. whether the chosen variance model has successfully removed the heteroscedasticity:

$$H_0 : \text{Var}(r_i) = \text{const} \quad \forall i \quad \text{vs.} \quad H_1 : \text{Var}(r_i) = h(\gamma_0 + \gamma_1 x_i). \quad (31)$$

The procedure regresses  $r_i^2$  on a constant and  $x_i$  by OLS. The LM statistic is

$$\text{LM} = n R^2, \quad (32)$$

where  $R^2$  is the coefficient of determination of the auxiliary regression. Under  $H_0$ ,  $\text{LM} \sim \chi_1^2$  asymptotically. A  $p$ -value below 0.05 indicates that residual spread still varies with  $x_i$ ; this suggests a more flexible variance model may be needed, or that a variance-stabilising transformation (e.g.  $\log y$  or  $\sqrt{y}$ ) should be applied before fitting.

### 9.3 Practical Guidance

Based on the diagnostics the application presents the following advice:

- **Curved or trending residuals:** try a different model (e.g. add a polynomial term, switch to an exponential or sigmoid function). The chosen functional form does not adequately capture the calibration relationship.

- **Fan-shaped spread in raw residuals:** enable or revise the **variance model** to let the model learn how noise scales with the mean response. Alternatively, apply a variance-stabilising transformation such as  $\log(y)$  or  $\sqrt{y}$  before fitting.
- **Significant Breusch–Pagan test ( $p < 0.05$ ) on standardised residuals:** the current variance model does not fully capture the heteroscedasticity; consider a different variance equation.
- **Random scatter around zero with non-significant test:** the model assumptions appear satisfied and inverse predictions can be trusted.

## 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 standard deviation given by the user-specified function  $\sigma_i = g(\mu_i, \sigma_y; \boldsymbol{\phi})$  (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.* Curved or systematic residuals suggest the model does not capture the true relationship; try a different equation.
- *Inadequate variance model.* The Breusch–Pagan test will flag remaining heteroscedasticity in the standardised residuals; revise the variance equation 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 work.
- *Correlated observations.* If measurements are time-dependent or spatially structured, the independence assumption is violated and intervals will be too narrow.

## 11 Implementation Notes

- **Equation parsing:** SymPy with implicit multiplication and `convert_xor` transformations. Both the mean function  $f(x; \boldsymbol{\theta})$  and the variance function  $g(\mu, \sigma_y; \boldsymbol{\phi})$  are parsed symbolically, enabling automatic inversion and differentiation.
- **Automatic differentiation:** PyTensor provides exact gradients for the NUTS sampler.
- **Sampling:** PyMC $\geq 5.10$  with the default NUTS implementation.
- **Diagnostics:** ArviZ for convergence statistics; statsmodels for the Breusch–Pagan test.
- **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. Shnайдман. 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.