CaliBR: Calibration with Bayesian inverse Regression

Mathematical Derivation and Assumptions

McClelland Lab, University College London

February 26, 2026

Contents

1 Introduction **1**

2 Notation **2**

3 The Calibration Model **2**

4 Heteroscedastic Variance Model **3**
4.1 Built-in Variance Functions . . . 3

5 Prior Distributions **3**
5.1 Log-Scale Parameterisation . . . 4

6 Posterior Distribution **4**

7 MCMC Sampling via NUTS **4**
7.1 Hamiltonian Monte Carlo . . . 4
7.2 The No-U-Turn Sampler . . . 5
7.3 Warm-up, Sampling, and Diagnostics . . . 5

8 Inverse Prediction **5**
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

9 Residual Diagnostics **7**
9.1 Residuals . . . 7
9.2 Breusch–Pagan Test . . . 7
9.3 Practical Guidance . . . 7

10 Summary of Assumptions **8**

11 Implementation Notes **8**

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
$\boldsymbol{x}^{\text{cal}} = (x_1, \ldots, x_n)^\top$	Calibration x-values
$\boldsymbol{y}^{\text{cal}} = (y_1, \ldots, y_n)^\top$	Calibration y-values
$f(x;\, \boldsymbol{\theta})$	User-specified calibration (mean) function
$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^\top$	Mean model parameters
σ_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
α	Tail probability for credible intervals ($\alpha = 0.025$ for 95% CI)

3 The Calibration Model

Two primary assumptions define the generative model:

Assumption 1 (Directionality of relationship between variables:)**.** 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 $f(x;\, a, b) = a\, e^{bx}$.

Assumption 2 (Structure of unexplained variance:)**.** Each observation equals the true mean response plus independent **Gaussian noise**. In the simplest (homoscedastic) case the noise has constant standard deviation σ_y:

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

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.

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

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

4 Heteroscedastic Variance Model

In many assays 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

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

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

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

and the heteroscedastic likelihood is

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

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	σ_y	Homoscedastic; $\boldsymbol{\phi} = \varnothing$
Proportional to mean	$\mu_i\,\sigma_y$	Constant coefficient of variation (CV)
Power-law	$\big\lvert\mu_i/A\big\rvert^{\gamma}\,\sigma_y$	A: geometric mean of $\boldsymbol{y}^{\text{cal}}$; $\boldsymbol{\phi} = \{\gamma\}$

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 $A = \exp\big(\frac{1}{n}\sum_{i=1}^{n}\log y_i\big)$ is a fixed scaling constant that makes σ_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 γ a uniform prior on $[0, 2]$:

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

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:

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

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

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, 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:

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

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,

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

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

6 Posterior Distribution

Applying Bayes' theorem:

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

where the marginal likelihood (evidence) is

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

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 $\boldsymbol{r} \in \mathbb{R}^d$ and defines the joint density

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

with *potential energy*

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

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 $\min\bigl(1,\,\exp(-\Delta H)\bigr)$, 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 $\{(\boldsymbol{\theta}^{(s)},\,\sigma_y^{(s)},\,\boldsymbol{\phi}^{(s)})\}_{s=1}^{N}$.
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_{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:

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

where the noise standard deviation is

$$\sigma^{*} \;=\; g\bigl(\mu^{*},\,\sigma_y;\,\boldsymbol{\phi}\bigr), \qquad \mu^{*} = f(x^{*};\,\boldsymbol{\theta}). \tag{21}$$

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

$$\sigma^{*} \;\approx\; g\bigl(y^{*},\,\sigma_y;\,\boldsymbol{\phi}\bigr). \tag{22}$$

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:

$$\boxed{p(x^* \mid y^*, \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}}) \;=\; \int p(x^* \mid y^*, \boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi})\;\; p(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi} \mid \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}})\, \mathrm{d}\boldsymbol{\theta}\, \mathrm{d}\sigma_y\, \mathrm{d}\boldsymbol{\phi}} \tag{23}$$

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

1. ***Compute noise scale:*** $\sigma^{*(s)} = g\big(y^*, \sigma_y^{(s)}; \boldsymbol{\phi}^{(s)}\big)$ *(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:*** $x^{*(s)} = f^{-1}\big(\tilde{y}^{(s)}; \boldsymbol{\theta}^{(s)}\big)$.

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. Fix a posterior draw $(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi})$. 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, $-\varepsilon^* \overset{d}{=} \varepsilon^*$, 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 $p(\boldsymbol{\theta}, \sigma_y, \boldsymbol{\phi} \mid \boldsymbol{y}^{\text{cal}}, \boldsymbol{x}^{\text{cal}})$, the composition produces samples from the marginal (23). □

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

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

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 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

$$\mathrm{CI}_{1-2\alpha} \;=\; \big[\, Q_\alpha,\;\; Q_{1-\alpha} \,\big], \tag{25}$$

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

$$\hat{x}_{\text{median}} = Q_{0.5}, \tag{26}$$

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

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

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 $\hat{\boldsymbol{\phi}}$:

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

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

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

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:

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

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

$$\text{LM} \;=\; n\, R^2, \tag{32}$$

where R^2 is the coefficient of determination of the auxiliary regression. Under H_0, LM $\sim \chi^2_1$ 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 (**??**).
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 ≥ 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. 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.