

# Laplace Approximation for NLME with Mutiple Groups

**Notation.** Let

$$y_{i,j,k} = \left( y_{i,j,k,1} \quad y_{i,j,k,2} \quad \cdots \quad y_{i,j,k,n_i} \right)'$$

be an  $n_i$ -dimensional random vector of outcome variables with support  $\mathbb{D}_y \subset \mathbb{R}^{n_i}$ . Let  $\eta_i$ ,  $\lambda_j$ , and  $a_k$  be  $n_\eta$ -,  $n_\lambda$ -, and  $n_a$ -dimensional random vectors of random effects with supports  $\mathbb{S}_\eta, \mathbb{S}_\lambda, \mathbb{S}_a$  defined over  $\mathbb{D}_\eta \subset \mathbb{R}^{n_\eta}$ ,  $\mathbb{D}_\lambda \subset \mathbb{R}^{n_\lambda}$ , and  $\mathbb{D}_a \subset \mathbb{R}^{n_a}$ , respectively.

The index  $i \in \{1, \dots, N\}$  denotes individuals (primary id). The index  $j \in \{1, \dots, J\}$  denotes a secondary grouping variable (e.g. schools or areas of living) such that  $j = j(i)$  is constant within an individual  $i$ , but a given  $j$  can be shared by multiple individuals. We further introduce a third grouping index  $k \in \{1, \dots, K\}$  (e.g. age group) such that  $k = k(i)$  is constant within an individual  $i$ , but *need not be constant within a  $j$ -group*; i.e., individuals sharing the same  $j$  may belong to different  $k$ -groups. (e.g. individuals sharing the same school must not be in the same age group.)

The collection of individual-level random effects for  $N$  individuals is denoted by

$$\eta = \left( \eta_1 \quad \eta_2 \quad \cdots \quad \eta_N \right)'$$

the collection of  $j$ -level random effects by

$$\lambda = \left( \lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_J \right)'$$

and the collection of  $k$ -level random effects by

$$a = \left( a_1 \quad a_2 \quad \cdots \quad a_K \right)'$$

Furthermore, let  $\theta \in \mathbb{D}_\theta \subset \mathbb{R}^{n_\theta}$  be a vector of fixed effects that parameterizes the distributional parameters

$$\phi_\eta = \Phi_\eta(\theta), \quad \phi_\lambda = \Phi_\lambda(\theta), \quad \phi_a = \Phi_a(\theta),$$

associated with the random-effects distributions of  $\eta_i$ ,  $\lambda_j$ , and  $a_k$ , respectively.

**Joint density.** The joint density of the outcome vectors and the random effects is denoted by

$$p\left( \{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}, \{\eta_i\}_{i \in \mathcal{C}}, \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}, \{a_k\}_{k \in \mathcal{K}(\mathcal{C})}; \theta \right),$$

where  $\theta$  indicates the density's dependency on the fixed effects. Here,  $\mathcal{C} \subseteq \{1, \dots, N\}$  is the (maximal) set of individuals that are mutually dependent through shared random effects, defined as the transitive closure of the relation

$$i \sim i' \iff j(i) = j(i') \text{ or } k(i) = k(i').$$

Equivalently,  $\mathcal{C}$  is the connected component (in the graph induced by shared group memberships) containing a reference individual  $i_0$ . Moreover,  $\mathcal{J}(\mathcal{C}) = \{j(i) : i \in \mathcal{C}\}$  and  $\mathcal{K}(\mathcal{C}) = \{k(i) : i \in \mathcal{C}\}$  denote the sets of  $j$ - and  $k$ -groups that occur among individuals in  $\mathcal{C}$ .

Define the product spaces

$$\mathbb{D}_{\eta, \mathcal{C}} := \prod_{i \in \mathcal{C}} \mathbb{D}_{\eta}, \quad \mathbb{D}_{\lambda, \mathcal{C}} := \prod_{j \in \mathcal{J}(\mathcal{C})} \mathbb{D}_{\lambda}, \quad \mathbb{D}_{a, \mathcal{C}} := \prod_{k \in \mathcal{K}(\mathcal{C})} \mathbb{D}_a.$$

The random effects can then be marginalized out of the joint density function. In particular, for a connected component (batch)  $\mathcal{C}$  we obtain the marginal likelihood contribution

$$\begin{aligned} p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}; \theta) &= \int_{\mathbb{D}_{\eta, \mathcal{C}}} \int_{\mathbb{D}_{\lambda, \mathcal{C}}} \int_{\mathbb{D}_{a, \mathcal{C}}} p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}} \mid \{\eta_i\}_{i \in \mathcal{C}}, \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}, \{a_k\}_{k \in \mathcal{K}(\mathcal{C})}; \theta) \\ &\quad \times p(\{\eta_i\}_{i \in \mathcal{C}}; \theta) p(\{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}; \theta) p(\{a_k\}_{k \in \mathcal{K}(\mathcal{C})}; \theta) da d\lambda d\eta. \end{aligned}$$

where  $d\eta = \prod_{i \in \mathcal{C}} d\eta_i$ ,  $d\lambda = \prod_{j \in \mathcal{J}(\mathcal{C})} d\lambda_j$ , and  $da = \prod_{k \in \mathcal{K}(\mathcal{C})} da_k$ .

Assuming conditional independence across individuals given the random effects, the conditional likelihood factorizes as

$$p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}} \mid \{\eta_i\}_{i \in \mathcal{C}}, \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}, \{a_k\}_{k \in \mathcal{K}(\mathcal{C})}; \theta) = \prod_{i \in \mathcal{C}} p(y_{i,j(i),k(i)} \mid \eta_i, \lambda_{j(i)}, a_{k(i)}; \theta).$$

Moreover, assuming independence of the random-effects groups  $\eta$ ,  $\lambda$ , and  $a$ , and independence within each group across indices, we have

$$p(\{\eta_i\}_{i \in \mathcal{C}}; \theta) = \prod_{i \in \mathcal{C}} p(\eta_i; \theta), \quad p(\{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}; \theta) = \prod_{j \in \mathcal{J}(\mathcal{C})} p(\lambda_j; \theta), \quad p(\{a_k\}_{k \in \mathcal{K}(\mathcal{C})}; \theta) = \prod_{k \in \mathcal{K}(\mathcal{C})} p(a_k; \theta).$$

Combining these yields

$$\begin{aligned} p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}; \theta) &= \int_{\mathbb{D}_{\eta, \mathcal{C}}} \int_{\mathbb{D}_{\lambda, \mathcal{C}}} \int_{\mathbb{D}_{a, \mathcal{C}}} \left[ \prod_{i \in \mathcal{C}} p(y_{i,j(i),k(i)} \mid \eta_i, \lambda_{j(i)}, a_{k(i)}; \theta) \right] \\ &\quad \times \left[ \prod_{i \in \mathcal{C}} p(\eta_i; \theta) \right] \left[ \prod_{j \in \mathcal{J}(\mathcal{C})} p(\lambda_j; \theta) \right] \left[ \prod_{k \in \mathcal{K}(\mathcal{C})} p(a_k; \theta) \right] da d\lambda d\eta. \end{aligned}$$

## Method: Laplace

**Taylor approximation.** For a connected component (batch)  $\mathcal{C}$ , define the stacked random-effects vector

$$b_{\mathcal{C}} := \begin{pmatrix} \{\eta_i\}_{i \in \mathcal{C}} \\ \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})} \\ \{a_k\}_{k \in \mathcal{K}(\mathcal{C})} \end{pmatrix} \in \mathbb{D}_{\mathcal{C}} := \mathbb{D}_{\eta, \mathcal{C}} \times \mathbb{D}_{\lambda, \mathcal{C}} \times \mathbb{D}_{a, \mathcal{C}}, \quad n_{\mathcal{C}} := \dim(b_{\mathcal{C}}) = |\mathcal{C}| n_{\eta} + |\mathcal{J}(\mathcal{C})| n_{\lambda} + |\mathcal{K}(\mathcal{C})| n_a.$$

To reduce notational burden, define the integrand

$$f_{\mathcal{C}}(b_{\mathcal{C}}; \theta) := \left[ \prod_{i \in \mathcal{C}} p(y_{i,j(i),k(i)} | \eta_i, \lambda_{j(i)}, a_{k(i)}; \theta) \right] \left[ \prod_{i \in \mathcal{C}} p(\eta_i; \theta) \right] \left[ \prod_{j \in \mathcal{J}(\mathcal{C})} p(\lambda_j; \theta) \right] \left[ \prod_{k \in \mathcal{K}(\mathcal{C})} p(a_k; \theta) \right],$$

so that

$$p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}; \theta) = \int_{\mathbb{D}_{\mathcal{C}}} f_{\mathcal{C}}(b_{\mathcal{C}}; \theta) db_{\mathcal{C}}.$$

Assuming  $f_{\mathcal{C}}(b_{\mathcal{C}}; \theta) > 0$  for all  $b_{\mathcal{C}} \in \mathbb{D}_{\mathcal{C}}$ , Laplace's method applies to the log-integrand

$$\ell_{\mathcal{C}}(b_{\mathcal{C}}; \theta) := \ln f_{\mathcal{C}}(b_{\mathcal{C}}; \theta).$$

Let  $b_{\mathcal{C}}^*$  denote the mode of  $\ell_{\mathcal{C}}(\cdot; \theta)$  (equivalently of  $f_{\mathcal{C}}$ ). The second-order Taylor polynomial of  $\ell_{\mathcal{C}}$  around  $b_{\mathcal{C}}^*$  is

$$T_{\mathcal{C}}(b_{\mathcal{C}}) = \ell_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) + \underbrace{\frac{\partial \ell_{\mathcal{C}}}{\partial b_{\mathcal{C}}} \Big|_{b_{\mathcal{C}}^*}}_{:= g_{\mathcal{C}}(b_{\mathcal{C}}^*)} (b_{\mathcal{C}} - b_{\mathcal{C}}^*) + \frac{1}{2} (b_{\mathcal{C}} - b_{\mathcal{C}}^*)' \underbrace{\frac{\partial^2 \ell_{\mathcal{C}}}{\partial b_{\mathcal{C}} \partial b_{\mathcal{C}}} \Big|_{b_{\mathcal{C}}^*}}_{:= H_{\mathcal{C}}(b_{\mathcal{C}}^*)} (b_{\mathcal{C}} - b_{\mathcal{C}}^*). \quad (1)$$

Using this approximation,

$$\begin{aligned} \int_{\mathbb{D}_{\mathcal{C}}} f_{\mathcal{C}}(b_{\mathcal{C}}; \theta) db_{\mathcal{C}} &= \int_{\mathbb{D}_{\mathcal{C}}} \exp(\ell_{\mathcal{C}}(b_{\mathcal{C}}; \theta)) db_{\mathcal{C}} \approx \int_{\mathbb{D}_{\mathcal{C}}} \exp(T_{\mathcal{C}}(b_{\mathcal{C}})) db_{\mathcal{C}} \\ &= f_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) \int_{\mathbb{D}_{\mathcal{C}}} \exp(g_{\mathcal{C}}(b_{\mathcal{C}}^*)'(b_{\mathcal{C}} - b_{\mathcal{C}}^*) + \frac{1}{2} (b_{\mathcal{C}} - b_{\mathcal{C}}^*)' H_{\mathcal{C}}(b_{\mathcal{C}}^*) (b_{\mathcal{C}} - b_{\mathcal{C}}^*)) db_{\mathcal{C}}. \end{aligned} \quad (2)$$

If  $b_{\mathcal{C}}^*$  is an interior mode of  $\mathbb{D}_{\mathcal{C}}$ , then  $g_{\mathcal{C}}(b_{\mathcal{C}}^*) = 0$ , and the integral simplifies to

$$\int_{\mathbb{D}_{\mathcal{C}}} f_{\mathcal{C}}(b_{\mathcal{C}}; \theta) db_{\mathcal{C}} \approx f_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) (2\pi)^{\frac{n_{\mathcal{C}}}{2}} \det(-H_{\mathcal{C}}(b_{\mathcal{C}}^*)^{-1})^{\frac{1}{2}}, \quad (3)$$

where the approximation is exact if  $\mathbb{D}_{\mathcal{C}} = \mathbb{R}^{n_{\mathcal{C}}}$ . Note that  $H_{\mathcal{C}}(b_{\mathcal{C}}^*)$  is negative definite at a strict local maximum, hence  $-H_{\mathcal{C}}(b_{\mathcal{C}}^*)$  is positive definite and the determinant is well-defined.

**Empirical Bayes estimate.** The empirical Bayes (mode) estimate for component (batch)  $\mathcal{C}$  is

$$b_{\mathcal{C}}^* = \arg \max_{b_{\mathcal{C}} \in \mathbb{D}_{\mathcal{C}}} \ell_{\mathcal{C}}(b_{\mathcal{C}}; \theta) = \arg \max_{b_{\mathcal{C}} \in \mathbb{D}_{\mathcal{C}}} \ln f_{\mathcal{C}}(b_{\mathcal{C}}; \theta).$$

**Laplace log-likelihood.** The Laplace approximation yields the following approximated log-likelihood contribution of component (batch)  $\mathcal{C}$ :

$$\begin{aligned} \ln p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}; \theta) &\approx \ln f_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) + \frac{n_{\mathcal{C}}}{2} \ln(2\pi) + \frac{1}{2} \ln \det(-H_{\mathcal{C}}(b_{\mathcal{C}}^*)^{-1}) \\ &= \ln f_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) + \frac{n_{\mathcal{C}}}{2} \ln(2\pi) - \frac{1}{2} \ln \det(-H_{\mathcal{C}}(b_{\mathcal{C}}^*)). \end{aligned} \quad (4)$$

**Aggregated Laplace log-likelihood.** Let  $\{\mathcal{C}_1, \dots, \mathcal{C}_M\}$ , with  $M \leq N$ , denote the partition of  $\{1, \dots, N\}$  into connected components (batches) induced by shared random effects. If the random effect is only on the individuals  $M = N$ . Under the conditional independence assumptions, the marginal likelihood factorizes across components, and the fully approximated log-likelihood is obtained by summing the component-wise Laplace contributions:

$$\begin{aligned} \ln p(\{y_{i,j(i),k(i)}\}_{i=1}^N; \theta) &= \sum_{m=1}^M \ln p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}_m}; \theta) \\ &\approx \sum_{m=1}^M \left[ \ln f_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*; \theta) + \frac{n_{\mathcal{C}_m}}{2} \ln(2\pi) - \frac{1}{2} \ln \det(-H_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*)) \right]. \end{aligned} \quad (5)$$

**Gradient w.r.t. fixed effects.** Let  $\{\mathcal{C}_1, \dots, \mathcal{C}_M\}$  denote the connected components (batches) and recall the aggregated Laplace log-likelihood

$$\tilde{\ell}(\theta) := \sum_{m=1}^M \left[ \ln f_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*; \theta) + \frac{n_{\mathcal{C}_m}}{2} \ln(2\pi) - \frac{1}{2} \ln \det(-H_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*)) \right],$$

where  $b_{\mathcal{C}_m}^*(\theta) = \arg \max_{b_{\mathcal{C}_m} \in \mathbb{D}_{\mathcal{C}_m}} \ln f_{\mathcal{C}_m}(b_{\mathcal{C}_m}; \theta)$  is the empirical Bayes (mode) estimate and  $H_{\mathcal{C}_m}(b; \theta) = \frac{\partial^2}{\partial b \partial b'} \ln f_{\mathcal{C}_m}(b; \theta)$ .

The gradient of  $\tilde{\ell}(\theta)$  is

$$\nabla_{\theta} \tilde{\ell}(\theta) = \sum_{m=1}^M \left[ \nabla_{\theta} \ln f_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*(\theta); \theta) - \frac{1}{2} \nabla_{\theta} \ln \det(-H_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*(\theta); \theta)) \right]. \quad (6)$$

Importantly,  $b_{\mathcal{C}_m}^*$  depends on  $\theta$ . By the envelope theorem applied to  $\ln f_{\mathcal{C}_m}(b; \theta)$  at its maximizer  $b = b_{\mathcal{C}_m}^*(\theta)$ , the derivative of the first term does not require differentiating through  $b_{\mathcal{C}_m}^*$ :

$$\nabla_{\theta} \ln f_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*(\theta); \theta) = \left. \frac{\partial}{\partial \theta} \ln f_{\mathcal{C}_m}(b; \theta) \right|_{b=b_{\mathcal{C}_m}^*(\theta)}. \quad (7)$$

For the curvature correction term, however, the dependence  $b_{\mathcal{C}_m}^*(\theta)$  must be taken into account:

$$\nabla_{\theta} \ln \det(-H_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*(\theta); \theta)) = \left. \frac{\partial}{\partial \theta} \ln \det(-H_{\mathcal{C}_m}(b; \theta)) \right|_{b=b_{\mathcal{C}_m}^*} + \left. \frac{\partial}{\partial b} \ln \det(-H_{\mathcal{C}_m}(b; \theta)) \right|_{b=b_{\mathcal{C}_m}^*} \frac{\partial b_{\mathcal{C}_m}^*(\theta)}{\partial \theta}. \quad (8)$$

The sensitivity  $\frac{\partial b_{\mathcal{C}_m}^*(\theta)}{\partial \theta}$  follows from implicit differentiation of the first-order optimality condition

$$g_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*(\theta); \theta) := \left. \frac{\partial}{\partial b} \ln f_{\mathcal{C}_m}(b; \theta) \right|_{b=b_{\mathcal{C}_m}^*(\theta)} = 0,$$

which yields

$$\frac{\partial b_{\mathcal{C}_m}^*(\theta)}{\partial \theta} = -H_{\mathcal{C}_m}(b_{\mathcal{C}_m}^*; \theta)^{-1} \left. \frac{\partial g_{\mathcal{C}_m}(b; \theta)}{\partial \theta} \right|_{b=b_{\mathcal{C}_m}^*}. \quad (9)$$

Finally, using the identity  $\nabla_X \ln \det(X) = X^{-\top}$  for invertible  $X$  and the differential  $d \ln \det(X) = \text{tr}(X^{-1}dX)$ , one convenient representation of the partial derivatives in (8) is

$$\begin{aligned} \frac{\partial}{\partial \theta} \ln \det(-H_{C_m}(b; \theta)) &= \text{tr} \left( [-H_{C_m}(b; \theta)]^{-1} \left[ -\frac{\partial H_{C_m}(b; \theta)}{\partial \theta} \right] \right), \\ \frac{\partial}{\partial b} \ln \det(-H_{C_m}(b; \theta)) &= \text{tr} \left( [-H_{C_m}(b; \theta)]^{-1} \left[ -\frac{\partial H_{C_m}(b; \theta)}{\partial b} \right] \right), \end{aligned} \quad (10)$$

where  $\frac{\partial H}{\partial \theta}$  and  $\frac{\partial H}{\partial b}$  are understood element-wise (yielding third-order derivative tensors).

**Gradient computation via automatic differentiation.** Direct application of automatic differentiation (AD) to the aggregated Laplace log-likelihood (5) would, in principle, require differentiating through the empirical Bayes solution  $b_C^*(\theta)$ , leading to deeply nested AD (optimization inside differentiation). This is computationally expensive and numerically fragile for large components. Instead, we exploit the structure of the Laplace approximation to minimize the degree of nesting.

*Envelope term.* For the leading term  $\ln f_C(b_C^*; \theta)$ , the envelope theorem applies, since  $b_C^*(\theta)$  maximizes  $\ln f_C(b; \theta)$  for fixed  $\theta$ . As a consequence,

$$\nabla_{\theta} \ln f_C(b_C^*; \theta) = \left. \frac{\partial}{\partial \theta} \ln f_C(b; \theta) \right|_{b=b_C^*},$$

and no differentiation through the optimizer is required. In practice, this term can be obtained using standard forward-mode AD applied to  $\ln f_C(b; \theta)$  with  $b$  treated as constant (after the EBE estimation).

*Curvature correction.* The remaining term,

$$-\frac{1}{2} \ln \det(-H_C(b_C^*; \theta)),$$

does depend on  $\theta$  both explicitly and implicitly through  $b_C^*(\theta)$ . However, this dependence can be handled without differentiating through the optimization algorithm. Instead, we proceed in three steps:

1. Compute the Hessian  $H_C(b_C^*; \theta) = \left. \frac{\partial^2}{\partial b \partial b'} \ln f_C(b; \theta) \right|_{b=b_C^*}$  using AD applied once to the log-integrand.
2. Obtain the sensitivity  $\frac{\partial b_C^*(\theta)}{\partial \theta}$  by implicit differentiation of the first-order optimality condition  $g_C(b_C^*; \theta) = 0$ , yielding

$$\frac{\partial b_C^*}{\partial \theta} = -H_C(b_C^*; \theta)^{-1} \left. \frac{\partial g_C(b; \theta)}{\partial \theta} \right|_{b=b_C^*}.$$

This step requires solving a linear system but avoids higher-order nested AD.

3. Evaluate the gradient of the log-determinant using matrix calculus identities:

$$\nabla_{\theta} \ln \det(-H) = \text{tr}((-H)^{-1}(-\nabla_{\theta}H)) + \text{tr}\left((-H)^{-1}(-\nabla_b H) \frac{\partial b^*}{\partial \theta}\right).$$

All partial derivatives of  $H$  are computed using AD with respect to either  $b$  or  $\theta$ , but never through the optimization loop itself.

*Resulting AD structure.* Overall, this strategy reduces the computation of  $\nabla_{\theta} \tilde{\ell}(\theta)$  to:

- one optimization per component to obtain  $b_{\mathcal{C}}^*$ ,
- first- and second-order AD of  $\ln f_{\mathcal{C}}(b; \theta)$ ,
- linear solves involving the Hessian  $H_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta)$ .

Crucially, no differentiation through the optimizer is required, and nested AD is avoided entirely. This yields a scalable and numerically stable procedure for gradient-based optimization of the Laplace-approximated likelihood.

**Optimizing Laplace evaluations via caching of empirical Bayes estimates.** Evaluating the Laplace-approximated objective  $\tilde{\ell}(\theta)$  and its gradient  $\nabla_{\theta} \tilde{\ell}(\theta)$  requires the empirical Bayes estimate (EBE)  $b_{\mathcal{C}}^*(\theta)$  for each component  $\mathcal{C}$ . Since the EBE is obtained by solving an inner optimization problem, recomputing  $b_{\mathcal{C}}^*(\theta)$  separately in every function evaluation and again in every gradient evaluation is unnecessarily expensive.

To avoid redundant inner solves, we cache the EBEs together with the fixed-effects vector at which they were computed. Concretely, for each component  $\mathcal{C}$  we store

$$(\theta_{\text{cache}}, \{b_{\mathcal{C}, \text{cache}}^*\}_{\mathcal{C}}), \quad \text{where } b_{\mathcal{C}, \text{cache}}^* = b_{\mathcal{C}}^*(\theta_{\text{cache}}).$$

At the beginning of an objective (or gradient) evaluation at  $\theta$ , we check whether  $\theta$  matches the cached value  $\theta_{\text{cache}}$ . If the fixed effects are identical, i.e.

$$\theta = \theta_{\text{cache}},$$

we reuse the stored EBEs  $b_{\mathcal{C}, \text{cache}}^*$  for all components. Otherwise, we recompute  $b_{\mathcal{C}}^*(\theta)$ , update the cache by setting  $\theta_{\text{cache}} \leftarrow \theta$  and  $b_{\mathcal{C}, \text{cache}}^* \leftarrow b_{\mathcal{C}}^*(\theta)$ , and then proceed with the Laplace evaluation.

This caching strategy ensures that, for a given  $\theta$ , the expensive computation of  $b_{\mathcal{C}}^*(\theta)$  is performed at most once, even if the optimizer requests both the objective and gradient at the same iterate. In practice, this substantially reduces runtime for gradient-based optimization routines that evaluate  $\tilde{\ell}(\theta)$  and  $\nabla_{\theta} \tilde{\ell}(\theta)$  sequentially at identical parameter values.

**Hessian w.r.t. fixed effects for inference (inverse Hessian).** For inference based on the Laplace-approximated marginal log-likelihood, we require the observed information  $\mathcal{I}(\theta) \approx -\nabla_{\theta}^2 \tilde{\ell}(\theta)$  and, in particular, the inverse Hessian  $\left[-\nabla_{\theta}^2 \tilde{\ell}(\theta)\right]^{-1}$ . We therefore compute the exact Hessian of the Laplace objective  $\tilde{\ell}(\theta)$  (up to the Laplace approximation itself) while avoiding differentiation through the EBE solver.

*Component-wise decomposition.* For each component  $\mathcal{C}$ , define

$$\ell_{\mathcal{C}}(b; \theta) := \ln f_{\mathcal{C}}(b; \theta), \quad g_{\mathcal{C}}(b; \theta) := \nabla_b \ell_{\mathcal{C}}(b; \theta), \quad H_{\mathcal{C}}(b; \theta) := \nabla_b^2 \ell_{\mathcal{C}}(b; \theta),$$

and let  $b_{\mathcal{C}}^*(\theta)$  satisfy  $g_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) = 0$ . The Laplace log-likelihood contribution is

$$\tilde{\ell}_{\mathcal{C}}(\theta) = \ell_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) + \frac{n_{\mathcal{C}}}{2} \ln(2\pi) - \frac{1}{2} \ln \det(-H_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta)), \quad \tilde{\ell}(\theta) = \sum_{m=1}^M \tilde{\ell}_{\mathcal{C}_m}(\theta).$$

Hence,

$$\nabla_{\theta}^2 \tilde{\ell}(\theta) = \sum_{m=1}^M \nabla_{\theta}^2 \tilde{\ell}_{\mathcal{C}_m}(\theta).$$

*EBE sensitivity (no differentiation through the optimizer).* Let

$$G_{\mathcal{C}}(b; \theta) := \nabla_{\theta} g_{\mathcal{C}}(b; \theta) = \nabla_{\theta} \nabla_b \ell_{\mathcal{C}}(b; \theta) \in \mathbb{R}^{n_{\mathcal{C}} \times n_{\theta}}.$$

Implicit differentiation of  $g_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) = 0$  gives

$$\frac{\partial b_{\mathcal{C}}^*}{\partial \theta} = -H_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta)^{-1} G_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta), \quad (11)$$

which requires only linear solves with  $H_{\mathcal{C}}$ .

*Hessian of the envelope term (exact, second order only).* Differentiating the envelope identity yields the Schur-complement form

$$\nabla_{\theta}^2 \ell_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta) = \nabla_{\theta}^2 \ell_{\mathcal{C}}(b; \theta) \Big|_{b=b_{\mathcal{C}}^*} - G_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta)^{\top} H_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta)^{-1} G_{\mathcal{C}}(b_{\mathcal{C}}^*; \theta). \quad (12)$$

This avoids nested AD entirely: it needs  $\nabla_{\theta}^2 \ell_{\mathcal{C}}$ ,  $G_{\mathcal{C}}$ , and solves in  $H_{\mathcal{C}}$ , all evaluated at  $(b_{\mathcal{C}}^*, \theta)$ .

*Hessian of the curvature correction via directional derivatives (no explicit 4th-order tensors).*

Let

$$A_{\mathcal{C}}(\theta) := -H_{\mathcal{C}}(b_{\mathcal{C}}^*(\theta); \theta) \quad (\text{positive definite at the mode}), \quad c_{\mathcal{C}}(\theta) := -\frac{1}{2} \ln \det A_{\mathcal{C}}(\theta).$$

For any direction  $u \in \mathbb{R}^{n_{\theta}}$ , define the directional EBE sensitivity

$$\dot{b}_{\mathcal{C}}(u) := \frac{\partial b_{\mathcal{C}}^*}{\partial \theta} u = -H_{\mathcal{C}}^{-1} G_{\mathcal{C}} u,$$

and the corresponding directional change in the matrix  $A_C$ :

$$\dot{A}_C(u) = - \left( \frac{\partial H_C}{\partial \theta} [u] + \frac{\partial H_C}{\partial b} [\dot{b}_C(u)] \right), \quad (13)$$

where  $\frac{\partial H}{\partial \theta} [u]$  and  $\frac{\partial H}{\partial b} [\cdot]$  denote directional derivatives (not full tensors).

Then the gradient and Hessian of the curvature correction can be expressed using matrix differential identities:

$$\nabla_{\theta} c_C(\theta) u = -\frac{1}{2} \text{tr} \left( A_C^{-1} \dot{A}_C(u) \right),$$

and for two directions  $u, v \in \mathbb{R}^{n_\theta}$ ,

$$\nabla_{\theta}^2 c_C(\theta) [u, v] = -\frac{1}{2} \left( -\text{tr} \left( A_C^{-1} \dot{A}_C(u) A_C^{-1} \dot{A}_C(v) \right) + \text{tr} \left( A_C^{-1} \ddot{A}_C(u, v) \right) \right), \quad (14)$$

with  $\ddot{A}_C(u, v)$  the second directional derivative of  $A_C(\theta)$ . Importantly, both  $\dot{A}_C(u)$  and  $\ddot{A}_C(u, v)$  can be computed with AD using *directional* Jacobian-/Hessian-vector products of  $H_C(b; \theta)$  w.r.t.  $(b, \theta)$ , combined with the linear solve for  $\dot{b}_C(u)$ . This avoids forming third-/fourth-order derivative tensors explicitly.

*Assembling the full Hessian.* Combining (12) and (14), we obtain

$$\nabla_{\theta}^2 \tilde{\ell}_C(\theta) = \nabla_{\theta}^2 \ell_C(b_C^*; \theta) + \nabla_{\theta}^2 c_C(\theta).$$

In practice, we compute  $\nabla_{\theta}^2 \tilde{\ell}(\theta)$  by repeatedly evaluating Hessian-vector products  $\nabla_{\theta}^2 \tilde{\ell}(\theta) u$  (using the above directional formulas and AD for the required derivative products), and, if needed, constructing the full matrix by choosing  $u = e_r$  (standard basis vectors). The resulting Hessian can be factorized (e.g. Cholesky) to obtain the inverse Hessian for inference.

## Method: MCEM (MCMC-EM) as an alternative to Laplace

**Setup.** Recall that the marginal likelihood is obtained by integrating out the random effects. In the present notation, for each connected component (batch)  $\mathcal{C}$  we write the complete-data (joint) density as

$$p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}, b_C; \theta) = p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}} | b_C; \theta) p(b_C; \theta),$$

where  $b_C = (\{\eta_i\}_{i \in \mathcal{C}}, \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}, \{a_k\}_{k \in \mathcal{K}(\mathcal{C})})$ . Under the conditional independence assumptions used above,

$$\begin{aligned} p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}} | b_C; \theta) &= \prod_{i \in \mathcal{C}} p(y_{i,j(i),k(i)} | \eta_i, \lambda_{j(i)}, a_{k(i)}; \theta), \\ p(b_C; \theta) &= \prod_{i \in \mathcal{C}} p(\eta_i; \theta) \prod_{j \in \mathcal{J}(\mathcal{C})} p(\lambda_j; \theta) \prod_{k \in \mathcal{K}(\mathcal{C})} p(a_k; \theta). \end{aligned}$$

The marginal likelihood factorizes over batches  $\{\mathcal{C}_1, \dots, \mathcal{C}_M\}$ , hence the marginal log-likelihood is

$$\ell(\theta) = \sum_{m=1}^M \ln p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}_m}; \theta), \quad p(\{y\}_{i \in \mathcal{C}}; \theta) = \int_{\mathbb{D}_{\mathcal{C}}} p(\{y\}_{i \in \mathcal{C}}, b_{\mathcal{C}}; \theta) db_{\mathcal{C}}.$$

**EM principle.** EM maximizes  $\ell(\theta)$  by iterating between:

- **E-step:** compute the conditional expectation of the complete-data log-likelihood under the current parameter value  $\theta^{(t)}$ ,
- **M-step:** maximize that expected complete-data log-likelihood w.r.t.  $\theta$ .

Define the complete-data log-likelihood contribution of batch  $\mathcal{C}$  as

$$\ell_{\mathcal{C}}^c(b_{\mathcal{C}}; \theta) := \ln p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}, b_{\mathcal{C}}; \theta).$$

Then the EM auxiliary function is

$$Q(\theta \mid \theta^{(t)}) := \mathbb{E}_{b \mid y, \theta^{(t)}} [\ln p(y, b; \theta)] = \sum_{m=1}^M \mathbb{E}_{b_{\mathcal{C}_m} \mid \{y\}_{i \in \mathcal{C}_m}, \theta^{(t)}} [\ell_{\mathcal{C}_m}^c(b_{\mathcal{C}_m}; \theta)],$$

where the second equality uses that the posterior  $p(b \mid y, \theta^{(t)})$  factorizes over connected components, and thus the expectation decomposes over batches.

**MCEM E-step (MCMC approximation).** In general, the posterior

$$p(b_{\mathcal{C}} \mid \{y\}_{i \in \mathcal{C}}, \theta^{(t)}) \propto p(\{y\}_{i \in \mathcal{C}} \mid b_{\mathcal{C}}; \theta^{(t)}) p(b_{\mathcal{C}}; \theta^{(t)})$$

is not available in closed form. MCEM replaces the exact conditional expectation by a Monte Carlo average based on MCMC draws

$$b_{\mathcal{C}}^{(1)}, \dots, b_{\mathcal{C}}^{(S)} \sim p(b_{\mathcal{C}} \mid \{y\}_{i \in \mathcal{C}}, \theta^{(t)}),$$

generated e.g. by Metropolis–Hastings, HMC/NUTS, or Gibbs sampling when conditional distributions are available. The batch-wise contribution is approximated by

$$\widehat{Q}_{\mathcal{C}}(\theta \mid \theta^{(t)}) := \frac{1}{S} \sum_{s=1}^S \ell_{\mathcal{C}}^c(b_{\mathcal{C}}^{(s)}; \theta), \quad \widehat{Q}(\theta \mid \theta^{(t)}) = \sum_{m=1}^M \widehat{Q}_{\mathcal{C}_m}(\theta \mid \theta^{(t)}).$$

Under standard regularity and ergodicity conditions for the MCMC kernel,  $\widehat{Q}(\theta \mid \theta^{(t)}) \rightarrow Q(\theta \mid \theta^{(t)})$  as  $S \rightarrow \infty$ .

**M-step.** The M-step updates the parameters by maximizing the Monte Carlo approximation:

$$\theta^{(t+1)} = \arg \max_{\theta \in \mathbb{D}_{\theta}} \widehat{Q}(\theta \mid \theta^{(t)}).$$

When a closed-form maximizer is not available, a numerical optimizer can be used. In that case, gradients can be obtained by exchanging differentiation and summation:

$$\nabla_{\theta} \widehat{Q}(\theta \mid \theta^{(t)}) = \frac{1}{S} \sum_{m=1}^M \sum_{s=1}^S \nabla_{\theta} \ell_{\mathcal{C}_m}^c(b_{\mathcal{C}_m}^{(s)}; \theta),$$

where each term can be computed with AD because  $b_{\mathcal{C}_m}^{(s)}$  is treated as constant during the M-step (the sampling distribution depends on  $\theta^{(t)}$ , not on the  $\theta$  being optimized in the M-step). This avoids nested AD through the MCMC procedure.

**Practical considerations (rigorous convergence control).** Because  $\widehat{Q}$  is noisy for finite  $S$ , the classical MCEM strategy increases the number of MCMC samples with iterations, e.g.  $S = S_t \rightarrow \infty$ , to ensure that the Monte Carlo error diminishes as  $t$  grows. A common sufficient condition is that the Monte Carlo error decreases fast enough such that optimization error dominates (e.g. by increasing  $S_t$  over iterations). An alternative is stochastic approximation EM (SAEM), which updates  $Q$  via a Robbins–Monro recursion; SAEM typically uses a fixed MCMC cost per iteration and a decreasing step size to ensure convergence.

**Relation to Laplace.** Laplace approximates the batch integral by a local Gaussian expansion around the posterior mode  $b_{\mathcal{C}}^*(\theta)$ . In contrast, MCEM targets the same marginal likelihood but approximates the E-step expectation using samples from the full posterior  $p(b_{\mathcal{C}} \mid y, \theta^{(t)})$ , thereby avoiding local-Gaussian assumptions at the cost of MCMC computation.

## Method: SAEM (Stochastic Approximation EM) with block MCMC

**Setup.** As in the MCEM formulation, the complete-data log-likelihood contribution of a connected component (batch)  $\mathcal{C}$  is

$$\ell_{\mathcal{C}}^c(b_{\mathcal{C}}; \theta) := \ln p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}, b_{\mathcal{C}}; \theta), \quad \ell^c(b; \theta) = \sum_{m=1}^M \ell_{\mathcal{C}_m}^c(b_{\mathcal{C}_m}; \theta),$$

where  $b_{\mathcal{C}} = (\{\eta_i\}_{i \in \mathcal{C}}, \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}, \{a_k\}_{k \in \mathcal{K}(\mathcal{C})})$ . At iteration  $t$ , SAEM targets the batch-wise posterior

$$\pi_{\mathcal{C}}^{(t)}(b_{\mathcal{C}}) := p(b_{\mathcal{C}} \mid \{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}}, \theta^{(t)}) \propto p(\{y_{i,j(i),k(i)}\}_{i \in \mathcal{C}} \mid b_{\mathcal{C}}; \theta^{(t)}) p(b_{\mathcal{C}}; \theta^{(t)}).$$

**Block MCMC within batches.** Since dependence only propagates within connected components, it is natural to construct a Markov kernel that updates  $b_{\mathcal{C}}$  batch-wise. Concretely, for each  $\mathcal{C}$  we employ a block-transition kernel  $K_{\mathcal{C}}^{(t)}$  that leaves  $\pi_{\mathcal{C}}^{(t)}$  invariant and is implemented by successive updates of blocks, e.g.

$$b_{\mathcal{C}} = (\eta_{\mathcal{C}}, \lambda_{\mathcal{J}(\mathcal{C})}, a_{\mathcal{K}(\mathcal{C})}) \rightsquigarrow \eta_{\mathcal{C}} \rightsquigarrow \lambda_{\mathcal{J}(\mathcal{C})} \rightsquigarrow a_{\mathcal{K}(\mathcal{C})},$$

where  $\eta_{\mathcal{C}} = \{\eta_i\}_{i \in \mathcal{C}}$ ,  $\lambda_{\mathcal{J}(\mathcal{C})} = \{\lambda_j\}_{j \in \mathcal{J}(\mathcal{C})}$ , and  $a_{\mathcal{K}(\mathcal{C})} = \{a_k\}_{k \in \mathcal{K}(\mathcal{C})}$ . Each block update may be realized via Metropolis–Hastings, HMC, slice sampling, or Gibbs steps when available. This formulation allows non-Gaussian random-effects distributions  $p(\eta_i; \theta)$ ,  $p(\lambda_j; \theta)$ ,  $p(a_k; \theta)$  and non-conjugate observation models.

**SA (stochastic approximation) recursion.** SAEM replaces the exact E-step expectation by a stochastic approximation updated using MCMC draws. There are two equivalent presentations:

(i) *General recursion for the auxiliary function.* Let

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{b|y, \theta^{(t)}}[\ell^{\mathcal{C}}(b; \theta)] = \sum_{m=1}^M \mathbb{E}_{b_{\mathcal{C}_m} \sim \pi_{\mathcal{C}_m}^{(t)}}[\ell_{\mathcal{C}_m}^{\mathcal{C}}(b_{\mathcal{C}_m}; \theta)].$$

SAEM maintains a running approximation  $\widehat{Q}^{(t)}(\theta)$  and updates it as

$$\widehat{Q}^{(t+1)}(\theta) = (1 - \gamma_{t+1}) \widehat{Q}^{(t)}(\theta) + \gamma_{t+1} \ell^{\mathcal{C}}(b^{(t+1)}; \theta), \quad (15)$$

where  $b^{(t+1)}$  is obtained by MCMC, i.e.  $b_{\mathcal{C}}^{(t+1)} \sim K_{\mathcal{C}}^{(t)}(\cdot \mid b_{\mathcal{C}}^{(t)})$  for each batch  $\mathcal{C}$  (or a subset of batches, see below).

(ii) *Sufficient-statistics recursion (exponential-family case).* If the complete-data model admits a representation

$$\ell^{\mathcal{C}}(b; \theta) = \langle S(y, b), \psi(\theta) \rangle - A(\theta) + \text{const}(y, b),$$

for some sufficient statistics  $S(y, b)$ , then (15) can be replaced by tracking

$$s^{(t)} \approx \mathbb{E}_{b|y, \theta^{(t)}}[S(y, b)]$$

via the Robbins–Monro recursion

$$s^{(t+1)} = s^{(t)} + \gamma_{t+1} \left( S(y, b^{(t+1)}) - s^{(t)} \right). \quad (16)$$

The M-step can then often be expressed in closed form as a function of  $s^{(t+1)}$ . This path is preferred when available; otherwise, one reverts to the general recursion (15) with a numerical M-step.

**Step-size schedule.** The sequence  $(\gamma_t)_{t \geq 1}$  is chosen according to the canonical SAEM schedule

$$\gamma_t = \begin{cases} 1, & t \leq t_0, \\ (t - t_0)^{-\kappa}, & t > t_0, \end{cases} \quad \kappa \in (1/2, 1],$$

where  $t_0$  and  $\kappa$  are user-adjustable hyperparameters. The standard Robbins–Monro conditions,  $\sum_{t=1}^{\infty} \gamma_t = \infty$  and  $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ , hold for  $\kappa \in (1/2, 1]$ .

**M-step.** The parameter update is defined as

$$\theta^{(t+1)} \in \arg \max_{\theta \in \mathbb{D}_\theta} \widehat{Q}^{(t+1)}(\theta), \quad (17)$$

or, in the sufficient-statistics formulation,  $\theta^{(t+1)} = \arg \max_{\theta} Q(\theta \mid s^{(t+1)})$ . When (17) is solved numerically, gradients can be computed with AD by treating the current MCMC draw(s)  $b^{(t+1)}$  (or the current statistics  $s^{(t+1)}$ ) as fixed within the M-step; this avoids nested differentiation through the MCMC kernels.

**Batch-wise and parallel update schedules.** Since the posterior factorizes across connected components, SAEM can update batches independently. Let  $\mathcal{M}_t \subseteq \{1, \dots, M\}$  denote the set of batches updated at iteration  $t$ . A general user-configurable scheme is:

- For each  $m \in \mathcal{M}_t$ , generate one (or several) MCMC transition(s) using  $K_{\mathcal{C}_m}^{(t)}$  to obtain  $b_{\mathcal{C}_m}^{(t+1)}$ .
- For  $m \notin \mathcal{M}_t$ , set  $b_{\mathcal{C}_m}^{(t+1)} := b_{\mathcal{C}_m}^{(t)}$ .

This framework covers fully parallel SAEM ( $\mathcal{M}_t = \{1, \dots, M\}$  for all  $t$ ), minibatch SAEM (random  $\mathcal{M}_t$  of fixed size), and deterministic schedules. The stochastic approximation updates (15) or (16) then use the resulting global state  $b^{(t+1)}$  (or the corresponding batch contributions) to update  $\widehat{Q}$  or  $s$ .

**Remarks on rigor and convergence.** Under standard conditions—ergodicity of each batch kernel  $K_{\mathcal{C}}^{(t)}$  with invariant distribution  $\pi_{\mathcal{C}}^{(t)}$ , regularity of the complete-data likelihood, and Robbins–Monro step sizes ( $\gamma_t$ )—SAEM converges to a (local) maximizer of the observed-data likelihood. The batch-wise factorization is particularly advantageous: it reduces MCMC state dimension, improves mixing, and allows scalable parallel implementations without changing the underlying likelihood target.

**Exponential-family structure of the complete-data likelihood: refined classification.** The applicability of a sufficient-statistics SAEM formulation depends on whether the *complete-data log-likelihood*

$$\ln p(y, b; \theta) = \ln p(y \mid b; \theta) + \ln p(b; \theta)$$

is an exponential family *in the parameter*  $\theta$ . This is a stronger requirement than the observation model being an exponential family in  $y$  for fixed  $\theta$ . Below we classify common model combinations accordingly.

**Class A: Exponential family in  $\theta$  (closed-form or low-dimensional M-step).**

*A1. Gaussian outcomes with homoscedastic variance.*

$$y_{i,j,k} \mid b \sim \mathcal{N}(\mu_{i,j,k}(b, \theta), \sigma^2), \quad b \sim \mathcal{N}(0, \Omega),$$

with  $\theta = (\beta, \sigma^2, \Omega)$ . *Result:* Quadratic complete-data likelihood. Sufficient statistics exist for all variance components.

*A2. Gaussian outcomes with proportional (multiplicative) noise.*

$$y_{i,j,k} \mid b \sim \mathcal{N}(\mu_{i,j,k}(b, \theta), \sigma^2 v_{i,j,k}(b)^2),$$

where  $v_{i,j,k}(b)$  is known given  $b$  (e.g.  $v = \mu$ , log-normal error). *Result:* The dependence on  $\sigma^2$  enters only through  $\ln \sigma^2$  and  $1/\sigma^2$ ; the complete-data likelihood remains exponential family in  $\sigma^2$ .

*A3. GLMMs with canonical links and Gaussian random effects.*

- Bernoulli (logit / probit)
- Binomial (logit)
- Poisson (log)
- Gamma (inverse / log)

with Gaussian random effects. *Result:* Complete-data likelihood is exponential family in  $\theta$ , although the marginal likelihood is not.

*A4. Conjugate hierarchical models.*

- Poisson–Gamma
- Binomial–Beta
- Multinomial–Dirichlet

*Result:* Full conjugacy yields exponential-family structure and closed-form SAEM updates.

*A5. Survival models with Gamma frailty.*

$$\lambda(t \mid b) = b \lambda_0(t) \exp(X\beta), \quad b \sim \text{Gamma}(\alpha, \beta).$$

*Result:* Complete-data likelihood is exponential family in  $(\beta, \alpha, \beta)$ .

**Class B: Not exponential family in  $\theta$  (numerical M-step required).**

*B1. Gaussian outcomes with heteroscedastic variance depending nonlinearly on  $\theta$ .*

$$y \mid b \sim \mathcal{N}(\mu(b, \theta), \sigma^2(b, \theta)),$$

with, for example,

$$\sigma(b, \theta) = \sqrt{(a + c\mu(b, \theta))^2} \quad \text{or} \quad \sigma^2(b, \theta) = \exp(\alpha + \beta\mu(b, \theta)).$$

*Result:* The log-likelihood contains terms such as  $\ln(a + c\mu)$  and  $(a + c\mu)^{-2}$ , which cannot be written in exponential-family form in  $\theta$ . SAEM remains valid, but the M-step must be solved numerically.

*B2. Combined additive + proportional error models.*

$$\sigma^2(b, \theta) = \sigma_a^2 + \sigma_p^2 \mu(b, \theta)^2.$$

*Result:* Although common in NLME, this variance structure breaks exponential-family structure in  $(\sigma_a^2, \sigma_p^2)$ ; numerical M-step required.

*B3. Non-Gaussian, non-conjugate random effects.*

- Lognormal frailty
- Student- $t$  random effects
- Mixture distributions

*Result:* No sufficient-statistics form in  $\theta$ ; SAEM via general stochastic approximation.

*B4. Nonlinear mixed-effects models with structural parameters inside nonlinear predictors.*

$$y_i = f(t_i, \theta, \eta_i) + \varepsilon_i.$$

*Result:* Even with Gaussian noise, structural parameters inside  $f$  typically break exponential-family structure; only variance components may admit closed-form updates.

*B5. Models with truncation, censoring, or constraints depending on  $\theta$ .* *Result:* Normalizing constants depend on  $\theta$  in a non-linear way; numerical M-step required.

### Summary and implementation guidance.

- Exponential-family structure in  $\theta \Rightarrow$  SAEM with sufficient statistics and closed-form or low-dimensional M-step.
- Nonlinear dependence of  $\theta$  inside variances, link functions, or constraints  $\Rightarrow$  SAEM with numerical M-step.
- Both regimes share the same stochastic-approximation E-step and block MCMC sampler.