Process-based Bayesian Melding of Occupational Exposure Models and Industrial Workplace Data

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

João Vitor Dias Monteiro

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Doctor of Philosophy

Sudipto Banerjee

August, 2012

Acknowledgements

I wish to thank all people who made this thesis possible.

I have no words that can adequately express my gratitude to Professor Sudipto Banerjee. During the entire Ph.D. program, he always showed enthusiasm for being my advisor, helped me understand doubts that I had innumerous times, listened to my ideas (even when they were terrible) and provided encouragement. Quite often we had lunch together and we always had good conversation and good ideas. That really contributed to the existence of this thesis.

I would like to thank my committee members: Professors James Hodges, Wei Pan and Gurumurthy Ramachandran. During the qualifying exam last year, they gave great feedback and made me view this work from a different perspective.

I am also grateful to my friends in the Biostatistics department, especially Sunny Kim, who took several coffee breaks with me and provided comic relief. Also, thanks to the the staff of the Biostatistics department, especially Sally Olander and Megan Schlick, who were very helpful in solving administrative problems.

Lastly, and most importantly, I wish to thank my whole family: my wife Gayle, my mother Maria, my father Israel (in memory), my sisters Carolina and Soninha, my nephews João Gabriel and Miguel. Their support over the years was essential for me to finish the Ph.D. program.

Dedication

For Gayle Jacqueline Johnson: besides giving me emotional support, she helped me immensely with my many doubts with the English language.

Abstract

In industrial hygiene a worker's exposure to chemical, physical and biological agents is increasingly being modeled using deterministic physical models. However, predicting exposure in real workplace settings is challenging and approaches that simply regress on a physical model (e.g. straightforward non-linear regression) are less effective as they do not account for biases attributable, at least in part, to extraneous variability. This also impairs predictive performance. We recognize these limitations and provide a rich and flexible Bayesian hierarchical framework, which we call process-based Bayesian melding (PBBM), to synthesize the physical model with the field data. We reckon that the physical model, by itself, is inadequate for enhanced inferential performance and deploy (multivariate) Gaussian processes to capture extraneous uncertainties and underlying associations. We propose rich covariance structures for multiple outcomes using latent stochastic processes. We also pay attention to computational feasibility. In particular, we explore Markov chain Monte Carlo (MCMC) as well as Integrated Nested Laplace Approximation (INLA) to estimate PBBM parameters.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Importance and critical barriers

A key concern of industrial hygiene is the estimation of a worker's exposure to chemical, physical and biological agents. One way of exposure assessment proceeds from prediction of exposure through mathematical modeling that represents the physical processes generating chemical concentrations in the workplace. An accurate representation will deliver better concentration estimates and facilitate subsequent decision-making in exposure management. However, this is challenging because the workplace is usually notoriously complex and no physical model, or models, is likely to provide an adequate representation. It is, therefore, becoming increasingly clear that a synergy of physical and statistical models is needed to better estimate the processes in the workplace.

Physical models in industrial hygiene are typically based upon some simplifying assumptions about air-flow and contaminant transport pattern. For example, [1] have described a two-zone model for concentrations in a "near field" in the proximity of a source and a "far field" that is some distance away from the source (see Section 2.2). The resulting physical model is then described by ordinary differential equations (ODE's) that model the rate of change in concentrations using some physical parameters [2].

These parameters constitute inputs to the physical model and, in theory, determine when the system attains steady state. Frequently, there is uncertainty associated with them. A purely conceptual approach would assign "plausible" values to these inputs, perhaps from theoretical considerations [e.g., 3], and arrive at the steady state concentration. This steady state value is then used for subsequent exposure management. This approach fails to account for inherent uncertainties, which, unfortunately, can generate biased and artificially precise estimates for the steady state concentrations thereby skewing subsequent exposure management.

Improved inference can be achieved by using observations from the workplace. Concentration is usually measured over a finite set of time points. Depending upon the experiment, at each time point one could have concentration measurements at different distances from the source. For instance, the two-zone setting produces bivariate concentration measurements – one from the "near field" and another from the "far field". A plausible choice for inputs to the two-zone model could, perhaps, be obtained by training them using trial-and-error until satisfactory agreement between the output and concentration measurements is achieved. The approach, however, is unattractive. Not only could finding satisfactory agreement between the observations and the physical model's output be difficult, the procedure precludes quantification of uncertainty in estimation and prediction. Model assessment would be completely ad-hoc as well.

A more principled approach estimates the physical model's unknown inputs from the concentration measurements, in addition to prior information on the input parameters. Usually, some prior information regarding the inputs to the physical model is available. These are usually formed from physical considerations implied by the model or from experts with experience in workplace environments. A Bayesian modeling framework, which allows synthesis of information from different sources is, therefore, attractive.

In fact, the use of Bayesian statistics has gained popularity in occupational exposure. [4] presents the application of a Bayesian framework for retrospective exposure assessment of workers in a nickel smelter. [5] apply Bayesian statistical techniques to

the problem of classifying the exposure profile for a similar exposure group into one of the five exposure categories: 0, 1, 2, 3, or 4, corresponding to trivial (or very low) exposure, highly controlled, well controlled, controlled, and poorly controlled exposures. [6] developed an empirical hierarchical Bayesian approach that combines expert judgment with repeated measurements, physical model and an empirical model that make use of exposure determinants to provide a prior estimate of the exposure profile. [7] proposed a Bayesian framework to illustrate how a specific exposure model, i.e. the two-zone model, and information on model parameters together with knowledge of uncertainty and variability in these quantities can be used to not only provide better estimates of model outputs but also model parameters.

1.2 Bayesian melding

Synthesizing deterministic physical models with statistical models to achieve improved inference continues to garner attention. One approach, Bayesian melding [e.g., 8, 9, 10, 11, 12], achieves such synthesis by incorporating prior information on the inputs to the physical model, estimates them using their posterior distributions and carries out subsequent predictive inference. In its simplest form, Bayesian melding proceeds from a hierarchical model which regresses on the physical model. See, for example, [7] and [13] for two very different applications of this approach. We demonstrate, however, that straightforward Bayesian nonlinear regression can be highly ineffective in predicting exposure concentrations in industrial workplaces.

[10] and [14] recently applied spatial processes to meld information from monitoring sites with output from numerical models. They focused largely upon spatial interpolation and predictions using independent runs of the numerical model. Assessing uncertainty in model inputs was precluded by the complexity of the physical models therein. In different applications, [11, 12] propose Bayesian melding with single-outcome land use and transportation models. There is also related literature in the domain of "computer

models" [e.g., 15, 16]. This typically refers to settings where evaluating the numerical model is computationally onerous and a Gaussian process is used as a stochastic emulator or interpolator to approximate model outputs.

1.3 Goals

The current thesis aims to provide for industrial hygienists a Bayesian melding framework that accounts for extraneous variability in the field data that the physical models are unable to capture, and consequently help them to produce better exposure management. In addition, this thesis addresses two statistical issues that have received little attention in the Bayesian melding literature. First, we deal with associated multiple outcomes that are not only related by the physical model, but are also likely to produce associated residuals and correlated biases. Second, the data from industrial workplaces are, more often than not, *misaligned*. This means that concentration measurements from two different outcomes may not always have been observed at the same timepoint.

We deal with the above issues by deploying a multivariate stochastic process. Apart from modeling the usual residual variability, we show how the multivariate process can achieve the following important analytical objectives: (i) approximate the trend (or bias) missed by the physical model for concentrations in both fields, (ii) capture correlations across time (with process realizations acting as time-varying random effects), and (iii) model the correlations among the outcomes when we have multiple outcomes. This last objective is especially relevant when the exposure concentrations from the two fields are correlated.

The remainder of the thesis evolves as follows: in Chapter 2 we present an R package called **B2Z** that implements the Bayesian nonlinear regression (BNLR) by [7]. Here we show that for data from actual workplaces, BNLR may not work well. In Chapter 3 we introduce our process-based Bayesian melding approach (PBBM) and compare it to the BNLR through a simulation study and also using a two-zone field data set.

Chapter 4 shows how to perform computationally fast approximate inference for the PBBM. Finally, Chapter 5 concludes the thesis by presenting some discussion with an eye toward future work.

Chapter 2

B2Z: An R Package for Bayesian Two-Zone Models

2.1 Introduction

[7] proposed a Bayesian framework for estimating two-zone model input parameters and also predicting exposure concentrations in zones near and far away from the source of contamination. Their simulation study shows that the model was able to successfully estimate the two-zone model input parameters as well as predict the exposure concentrations at the near and fields over a period of time, under the condition that the data is generated from the model they were fitting.

In applied research, providing software with a proposed model encourages other researchers to explore the proposed model, detect potential issues, and advance methodological research. An exciting prospect in recent times that helps bring such sophisticated statistical methodology to users is the R project [17]. R is a language and environment for statistical computing and graphics that offers several built-in functions for mathematical computations. A convenient feature of R is the ability to create packages (libraries) that implement a new model. In addition, for computationally-intensive

tasks, C, C++ and Fortran programs can be linked and invoked by R at run time.

The present chapter introduces an R package called **B2Z** (`http://CRAN.R-project.org/package=B2Z`) that implements the Bayesian non-linear regression (BNLR) proposed by [7]. This package obtains random samples from the posterior distribution of the parameters and exposure concentrations for the BNLR. Currently, three different sampler algorithms are available to do such a task: sampling importance resampling, incremental mixture importance sampling, and the Metropolis-within-Gibbs sampler. In addition, the package also offers approximate Bayesian estimation using the Bayesian central limit theorem.

This chapter is organized as follows. In Section 2.2 we briefly describe the two-zone occupational exposure model. Section 2.3 recounts the BNLR framework. Section 2.4 briefly describes the sampler algorithms implemented in **B2Z**. Section 2.5 illustrates the use of **B2Z** with simulated and real data examples. Finally, Section 2.6 concludes the chapter with some discussion and thoughts.

2.2 Two-zone model

The two-zone (or two-component) model assumes the presence of a contamination source in the workplace and that the region very near and around the source is modeled as one well-mixed box, called the *near field*, while the rest of the room is another well-mixed box. This box is called the *far field* and there is some amount of air exchange between the two boxes.

Following convention, we assume that each field is a well mixed box, i.e., two distinct places that are in the same field have equal exposure concentration levels. Also, we assume that the contaminant's total mass is emitted at a constant rate G and that there is an airflow rate between the near field and far field equal to β. The final assumption is that there are supply and exhaust flow rates which are taken to be the same and equal to Q. Figure 2.1 is a schematic depiction of the dynamics of the system, where V_N and

V_F denote the volumes at the near and far field, respectively.

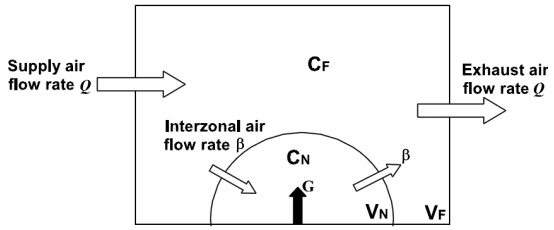


Figure 2.1: Dynamics of the two-zone model.

In this context, the hygienist models the exposure concentrations at the near and far fields based upon observations collected over a period of time. Figure 2.1, along with the assumptions, yields the following two-component model:

$$\frac{d}{dt}\mathbf{c}\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)=\mathbf{W}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)\mathbf{c}\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)+\mathbf{g}\left(\boldsymbol{\theta}_1;\mathbf{x}\right) \tag{2.1}$$

where $\mathbf{c}\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)=\begin{bmatrix} c_N\left(\boldsymbol{\theta}_1;\mathbf{x},t\right) \\ c_F\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)\end{bmatrix}$, $\mathbf{W}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)=\begin{bmatrix} -\frac{\beta}{V_N} & \frac{\beta}{V_N} \\ \frac{\beta}{V_F} & -\frac{(\beta+Q)}{V_F}\end{bmatrix}$, $\mathbf{g}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)=\begin{bmatrix} \frac{G}{V_N} \\ 0\end{bmatrix}$, $\boldsymbol{\theta}_1=\{\beta,Q,G\}$ and $\mathbf{x}=\{V_N,V_F\}$. The functions $c_N\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)$ and $c_F\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)$ are the exposure concentrations at time t in the near and far fields respectively.

The solution of (2.1) depends upon the eigenvalues of $\mathbf{W}(\boldsymbol{\theta}_1;\mathbf{x})$. When the eigenvalues are real and distinct, we obtain the following solution for (2.1):

$$\mathbf{c}(\boldsymbol{\theta}_1;\mathbf{x},t)=\exp\left(t\mathbf{W}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)\right)\mathbf{c}(\boldsymbol{\theta}_1;\mathbf{x},0)+\mathbf{W}^{-1}(\boldsymbol{\theta}_1;\mathbf{x})\left[\exp\left(t\mathbf{W}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)\right)-\mathbf{I}_2\right]\mathbf{g}(\boldsymbol{\theta}_1;\mathbf{x}), \tag{2.2}$$

where $\exp\left(t\mathbf{W}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)\right)$ is the matrix exponential (see Appendix A). Assuming that $c_N\left(\boldsymbol{\theta}_1;\mathbf{x},0\right)=c_F\left(\boldsymbol{\theta}_1;\mathbf{x},0\right)=0$, (2.2) can be simplified to produce the following unique

solution:

$$c_N\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)=\frac{G}{Q}+\frac{G}{\beta}+G\left(\frac{\beta Q+\lambda_2 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_1 t}-G\left(\frac{\beta Q+\lambda_1 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_2 t},$$

$$c_F\left(\boldsymbol{\theta}_1;\mathbf{x},t\right)=\frac{G}{Q}+G\left(\frac{\lambda_1 V_N+\beta}{\beta}\right)\left(\frac{\beta Q+\lambda_2 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_1 t}-G\left(\frac{\lambda_2 V_N+\beta}{\beta}\right)\left(\frac{\beta Q+\lambda_1 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_2 t}, \tag{2.3}$$

where λ_1 and λ_2 are the eigenvalues of $\mathbf{W}\left(\boldsymbol{\theta}_1;\mathbf{x}\right)$. These are available in closed form, and are negative real and distinct whenever β, Q, V_F and V_N are all strictly positive (see Appendix A). The latter two quantities are volumes of a chamber, hence positive. Physical considerations ensure that the same is true for β and Q. Assigning priors with positive support ensures a stable system with real solutions.

The exponential terms in (2.3) decay asymptotically to zero at large values of t. Consequently, the steady state solutions for the near and far fields are $G/Q+G/\beta$ and G/Q, respectively. Therefore, the model predicts a greater exposure intensity near the emission source compared to the one-compartment model in steady state conditions. Moreover, when β is less than or equal to Q, the steady state concentration in the far field is less than twice the steady state concentration in the near field. In general, Q increases relative to β as the room size increases. Thus, the model draws a distinction between exposures of workers near the source and those farther away from the source.

2.3 Bayesian non-linear regression for two-zone data

We describe in this section the BNLR for estimating model parameters and exposure concentrations in a two-zone model, which was summarized in the previous section. Notice from Eq. 2.3 that the solution of the system in (2.1) depends upon several parameters. Customarily, V_N and V_F are considered fixed and known, while β, Q and G are regarded as unknown parameters and will need to be estimated. Let $\mathbf{y}(t)=(y_1(t),y_2(t))^{\mathrm{T}}$ be a 2×1 vector corresponding to the natural logarithm of the exposure concentration at time point t, where the first and second entries of this vector are related

to the near and far fields, respectively. The observed value of $\mathbf{y}(t)$ is a combination of two components:

1. **Systematic component**: $\mathbf{c}(\boldsymbol{\theta}_1; \mathbf{x}, t) = (c_N(\boldsymbol{\theta}_1; \mathbf{x}, t), c_F(\boldsymbol{\theta}_1; \mathbf{x}, t))^{\mathrm{T}}$, the solution of the system of differential equations in (2.1) at time t;

2. **Measurement error process component**: $\boldsymbol{\epsilon}(t) = (\epsilon_1(t), \epsilon_2(t))^{\mathrm{T}}$, where $\epsilon_1(t)$ and $\epsilon_2(t)$ are the measurement error processes corresponding to the near and far field, respectively.

This leads to the following measurement model:

$$\mathbf{y}(t) = \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t) + \boldsymbol{\epsilon}(t), \tag{2.4}$$

where $\mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t) = (\log c_N(\boldsymbol{\theta}_1; \mathbf{x}, t), \log c_F(\boldsymbol{\theta}_1; \mathbf{x}, t))^{\mathrm{T}}$. Following [7], we assume Gaussian measurement error and, more specifically, the following two possibilities:

1. **Independent model**: $\boldsymbol{\epsilon}(t) \overset{iid}{\sim} \mathrm{N}_2\left(\mathbf{0}_2, \boldsymbol{\Sigma} = \begin{bmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{bmatrix}\right)$.

2. **Dependent model**: $\boldsymbol{\epsilon}(t) \overset{iid}{\sim} \mathrm{N}_2\left(\mathbf{0}_2, \boldsymbol{\Sigma} = \begin{bmatrix} \tau_1 & \tau_{12} \\ \tau_{12} & \tau_2 \end{bmatrix}\right)$.

In the independent model, the measurement errors at the near and far field are assumed to be uncorrelated, while the dependent model relaxes this assumption. For both models, it is assumed that the measurement errors across time are independent and identically distributed.

Let $\mathbf{y} = \left(\mathbf{y}^{\mathrm{T}}(t_1), \ldots, \mathbf{y}^{\mathrm{T}}(t_n)\right)^{\mathrm{T}}$ denote the $2n \times 1$ vector of observed log-concentrations from the near and far fields at n time points. Letting $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\Sigma}\}$ be the collection of unknown parameters, (2.4) and the assumptions made on the measurement errors produce the likelihood

$$p(\mathbf{y} \,|\, \boldsymbol{\theta}) \propto |\boldsymbol{\Sigma}|^{-\frac{n}{2}} \prod_{i=1}^{n} \exp\left\{-\frac{1}{2}\left(\mathbf{y}(t_i) - \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t_i)\right)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \left(\mathbf{y}(t_i) - \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t_i)\right)\right\}, \tag{2.5}$$

where $\mathbf{\Sigma}$ is the covariance matrix of the measurement error process. We assume that the components β, Q, G and $\mathbf{\Sigma}$ are independent a priori, so that the prior distribution for $\boldsymbol{\theta}$ is $p(\boldsymbol{\theta}) = p(\beta)p(Q)p(G)p(\mathbf{\Sigma})$. For the independent model, we assume that $p(\mathbf{\Sigma}) = p(\tau_1)p(\tau_2)$ with $\tau_1 \sim \text{IG}(a_1, b_1)$ and $\tau_2 \sim \text{IG}(a_2, b_2)$, where a_1 and a_2 are shape parameters and b_1 and b_2 are scale parameters for the inverse Gamma distribution. For the dependent model, $\mathbf{\Sigma} \sim \text{IW}(\mathbf{S}, v)$ where $\mathbf{S}$ is a scale matrix and v is the degrees of freedom for the inverse Wishart distribution. The parameterizations of the inverse gamma and inverse Wishart here are the same as in [18]. The parameters β, Q and G can have any prior distribution with positive support, i.e., they do not assign positive probabilities to any negative value. Based upon the above assumptions, the posterior distribution of $\boldsymbol{\theta}$ can be computed using Bayes rule as proportional to $p(\boldsymbol{\theta}) \times p(\mathbf{y} \,|\, \boldsymbol{\theta})$. However, the posterior distribution may not have a closed form precluding analytical inference. Our package **B2Z** has three different sampler algorithms available to obtain samples from the posterior distribution of $\boldsymbol{\theta}$. The algorithms are discussed in the next section.

2.4 Bayesian estimation

In this section we briefly discuss the three sampling algorithms and the approximation using the Bayesian central limit theorem that are available in our package **B2Z**. We also present some algorithmic implementation details.

2.4.1 Sampling importance resampling

Sampling importance resampling (SIR) [19, 20, 21] is a fairly straightforward algorithm used to obtain random samples from a probability distribution, here the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{y})$. Several variants of this algorithm exist [see, e.g., 22], but the basic idea is to sample $\boldsymbol{\theta}$ from an easily tractable distribution (e.g., the prior distribution) so that the SIR tends to choose $\boldsymbol{\theta}_i$'s corresponding to higher values of the likelihood. This

sampler is described in the following algorithm:

1. Obtain m i.i.d samples from the prior distribution $p(\boldsymbol{\theta})$. Denote each sample by $\boldsymbol{\theta}_i$, $i = 1, \ldots, m$;

2. For each sample $\boldsymbol{\theta}_i$, evaluate the likelihood $l_i = p(\mathbf{y} \,|\, \boldsymbol{\theta} = \boldsymbol{\theta}_i)$;

3. Compute the importance weights as:

$$w_i = \frac{l_i}{\sum_{k=1}^{m} l_k};$$

4. From the m samples obtained at the first step, select m samples (with replacement) using the weights w_i's.

In Step (3), the l_i's can be very close to zero so that a large proportion of the importance weights are close to zero as well. To assuage this issue, we implement in our package the following computational trick. We replace the computation of the importance weights in Step (3) for:

$$w_i = \frac{\exp(\widetilde{l}_i)}{\sum_{k=1}^{m} \exp(\widetilde{l}_k)};$$

where $\widetilde{l}_i = \log\left(p(\mathbf{y} \,|\, \boldsymbol{\theta} = \boldsymbol{\theta}_i)\right) + C$, and C is a large positive constant. While this may not fully resolve the issue of small weights, it does considerably increase the number of non-zero weights. Nevertheless, for the SIR to sample well from the posterior distribution, m must be large (thousands or even millions) which can be computationally expensive. In fact, in our examples, we often discovered the SIR to be returning very few distinct values, even with an m of size 50,000. This arises due to an inadequate exploration of the parameter domain. Also, it is important to the SIR that the prior distribution agrees with the likelihood. Otherwise very few distinct sampled points from the tails of the prior distribution have sizeable importance weights causing the final sample to

have few unique points. The incremental mixture importance sampling, described next, attempts to circumvent these problems.

2.4.2 Incremental mixture importance sampling

In contrast to the SIR, at each iteration the incremental mixture importance sampling (IMIS) [23, 24] adds samples from a multivariate normal distribution, centered at the point with the highest importance weight, to the current importance sampling distribution. This covers sections of the posterior distribution with high importance weights that are normally underrepresented by the importance sampling distribution. The IMIS algorithm is presented below:

1. Initial stage:

   (a) Draw N_0 i.i.d. samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_0}$ from the prior distribution of $\boldsymbol{\theta}$;

   (b) For each $\boldsymbol{\theta}_i$, evaluate the likelihood $l_i = p(\mathbf{y} \,|\, \boldsymbol{\theta} = \boldsymbol{\theta}_i)$ and compute its importance weight as:

   $$w_i^{(0)} = \frac{l_i}{\sum_{k=1}^{N_0} l_k};$$

   (c) Set $N_1 = N_0$.

2. Importance sampling stage: $k = 1$. **While** some stopping criterion (see below) is not satisfied **do**:

   (a) Denote by $\boldsymbol{\mu}^{(k)}$ the input with highest importance weight among the current importance sample up to iteration k;

   (b) Find the B inputs with smallest Mahalanobis distance to $\boldsymbol{\mu}^{(k)}$. The distances are calculated with respect to the prior covariance matrix of $\boldsymbol{\theta}$, denoted by $\mathbf{V}_{\boldsymbol{\theta}}$. More precisely, the Mahalanobis distance of an input $\mathbf{u}$ to $\boldsymbol{\mu}^{(k)}$ with

respect to $\mathbf{V}_{\boldsymbol{\theta}}$ is given by:

$$D = \sqrt{\left(\mathbf{u} - \boldsymbol{\mu}^{(k)}\right)^{\mathrm{T}} \mathbf{V}_{\boldsymbol{\theta}}^{-1} \left(\mathbf{u} - \boldsymbol{\mu}^{(k)}\right)}$$

(c) Denote by $u_1, \ldots, u_B$ the importance weights of the B inputs selected in the previous step;

(d) Denote by $\tilde{\boldsymbol{\Sigma}}^{(k)}$ the estimated weighted covariance matrix from the selected B inputs. The weight of the input j is given by:

$$v_j = \frac{(u_j + 1/N_k)}{\sum_{k=1}^{B} (u_j + 1/N_k)} \;\forall\; j = \{1, \ldots, B\};$$

(e) Draw B samples from a $\mathrm{N}_d\left(\boldsymbol{\mu}^{(k)}, \tilde{\boldsymbol{\Sigma}}^{(k)}\right)$, where d is the dimension of $\boldsymbol{\theta}$;

(f) Compute the likelihood for each new input from the previous step, and combine the new inputs with the previous ones;

(g) Update: $N_k = N_0 + B_k$;

(h) Compute the mixture sampling distribution $q^{(k)}$ at iteration k, given by:

$$q^{(k)}(\boldsymbol{\theta}_i) = \frac{N_0}{N_k} p(\boldsymbol{\theta}_i) + \frac{B}{N_k} \sum_{s=1}^{k} \mathrm{N}_d\left(\boldsymbol{\theta}_i \,|\, \boldsymbol{\mu}^{(s)}, \tilde{\boldsymbol{\Sigma}}^{(s)}\right), \forall\; i = \{1, 2, \ldots, N_k\}$$

where $p(\cdot)$ is the prior distribution of $\boldsymbol{\theta}$ and $\mathrm{N}_d(\cdot \,|\, \mathbf{m}, \mathbf{S})$ denotes the multivariate normal density with vector mean $\mathbf{m}$ and covariance matrix $\mathbf{S}$;

(i) Calculate the importance weights using the following formula:

$$w_i^{(k)} = c \times l_i \times \frac{p(\boldsymbol{\theta}_i)}{q^{(k)}(\boldsymbol{\theta}_i)} \;\forall\; i = \{1, 2, \ldots, N_k\}$$

where c is chosen so that the weights sum to 1;

(j) $k = k + 1$.

3. Resample stage: Once the stopping criterion (see below) at the importance sampling stage is satisfied, use the importance weights $w_1^{(K)}, \ldots, w_{N_K}^{(K)}$ to draw, with replacement, M inputs from the importance sample $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_K}$, where K is the total number of iterations at the importance sampling stage.

Stopping criterion: [24] suggest ending the importance sampling step when the expected fraction of unique points out of N_k is at least 0.632. **B2Z** follows this suggestion. However, the user can provide a maximum number of iterations at the importance sampling stage in case the stopping criterion takes too long to be met. [24] also suggest that a good choice for the input parameters is: $N_0 = 1000d$, $B = 100d$ and $M = 3000$. Recall that if the independent model is considered $d = 5$, otherwise $d = 6$.

2.4.3 Metropolis-within-Gibbs sampling

Gibbs sampling [25, 26] is a popular Markov chain Monte Carlo (MCMC) algorithm that samples from the full conditional distributions for each parameter. This is attractive in our context since the full conditional distributions for τ_1 and τ_2 in the independent model and for $\boldsymbol{\Sigma}$ in the dependent model are respectively given by:

1. **Independent model**:

$$\tau_1 \,|\, \boldsymbol{\theta}_1, \mathbf{y}, \mathbf{x} \sim \text{IG}\left(a_1 + \frac{n}{2},\, b_2 + \frac{1}{2}\sum_{j=1}^{n}(y_1(t_j) - \log(c_N(\boldsymbol{\theta}_1; \mathbf{x}, t_j)))^2\right),$$

$$\tau_2 \,|\, \boldsymbol{\theta}_1, \mathbf{y}, \mathbf{x} \sim \text{IG}\left(a_2 + \frac{n}{2},\, b_2 + \frac{1}{2}\sum_{j=1}^{n}(y_2(t_j) - \log(c_F(\boldsymbol{\theta}_1; \mathbf{x}, t_j)))^2\right).$$

2. **Dependent model**: $\boldsymbol{\Sigma} \,|\, \boldsymbol{\theta}_1, \mathbf{y} \sim \text{IW}(\mathbf{S}^*, v^*)$, where

   (a) $\mathbf{S}^* = \mathbf{S} + \sum_{j=1}^{n} \left(\mathbf{y}(t_j) - \log(\mathbf{c}(\boldsymbol{\theta}_1; \mathbf{x}, t_j))\right)\left(\mathbf{y}(t_j) - \log(\mathbf{c}(\boldsymbol{\theta}_1; \mathbf{x}, t_j))\right)^{\text{T}}$;

(b) $v^* = v + n$.

However, the full conditional distribution of $\boldsymbol{\theta}_1$ does not have a closed form and we sample from its full conditional distribution using the Metropolis algorithm (see Metropolis algorithm section below). This is called the Gibbs sampler with Metropolis step (or Metropolis-within-Gibbs). The algorithm is as follows:

Provide the initial value $\boldsymbol{\theta}_1^{(0)}$;
for k `in` $1 : N$ **do**
  Draw a sample from $\boldsymbol{\Sigma} \,|\, \boldsymbol{\theta}_1^{(k-1)}, \mathbf{y}$, and denote it as $\boldsymbol{\Sigma}^{(k)}$
  Using Metropolis sampler, draw a sample from $\boldsymbol{\theta}_1 \,|\, \boldsymbol{\Sigma}^{(k)}, \mathbf{y}$ and denote by $\boldsymbol{\theta}^{(k)}$.
end for

Metropolis algorithm

Metropolis [27, 28] is a well known MCMC sampling algorithm. Here, at each iteration we sample a candidate from a proposal distribution and then decide whether the candidate should be accepted or not. This decision is based on the ratio of the posterior distribution evaluated at the candidate and the previously accepted candidate. Since this is a ratio one needs to evaluate the posterior distribution only up to a proportionality constant. Several variants of the Metropolis sampler exist [see, e.g., 22]. The one that is currently implemented in **B2Z** is the *random-walk* Metropolis algorithm with normal proposals and is described as follows:

Provide the initial value $\boldsymbol{\theta}_1^{(0)}$;
for k `in` $1 : N$ **do**
  Generate a candidate $\boldsymbol{\theta}_1^{(*)}$ from a $\mathrm{N}_d\left(\boldsymbol{\theta}_1^{(k-1)}, \mathbf{V}\right)$
  $r = \dfrac{p\left(\boldsymbol{\theta}_1^{(*)} \,\middle|\, \mathbf{y}\right)}{p\left(\boldsymbol{\theta}_1^{(k-1)} \,\middle|\, \mathbf{y}\right)}$
  if $r \geq 1$ **then**
    $\boldsymbol{\theta}_1^{(k)} \leftarrow \boldsymbol{\theta}_1^{(*)}$

else

Generate a number u from a $\mathrm{U}(0,1)$

if $u < r$ **then**

$\boldsymbol{\theta}_1^{(k)} \leftarrow \boldsymbol{\theta}_1^{(*)}$

else

$\boldsymbol{\theta}_1^{(k)} \leftarrow \boldsymbol{\theta}_1^{(k-1)}$

end if

end if

end for

The input parameters for the Gibbs sampler with the Metropolis step algorithm are the number of updates N, the vector initial value $\boldsymbol{\theta}_1^{(0)}$, and a covariance matrix $\mathbf{V}$ for the proposal distribution. An approach that usually works well in practice estimates the posterior mode and uses it as an initial value. For $\mathbf{V}$, we use the negative inverse of the hessian matrix of the log posterior distribution evaluated at the posterior mode. This approach is implemented in **B2Z** as the default for setting initial values and specifying the proposal covariance matrix $\mathbf{V}$.

2.4.4 Bayesian central limit theorem

In contrast to the previous sections where we discussed sampler algorithms, in this section we briefly discuss the Bayesian central limit theorem (BCLT). This theorem states that under some assumptions we can use a Gaussian approximation to the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{y}) = \dfrac{f(\boldsymbol{\theta})}{\int f(\boldsymbol{\theta}) d\boldsymbol{\theta}}$, where $f(\boldsymbol{\theta}) = p(\mathbf{y} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta})$.

Consider a Taylor expansion of $\ln(f(\boldsymbol{\theta}))$ centered on the posterior mode $\boldsymbol{\theta}_0$. At $\boldsymbol{\theta}_0$ the gradient $\nabla f(\theta)$ will vanish. Thus the expansion around $\boldsymbol{\theta}_0$ is given by

$$\ln(f(\boldsymbol{\theta})) \simeq \ln(f(\boldsymbol{\theta}_0)) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^{\mathrm{T}} \mathbf{H} (\boldsymbol{\theta} - \boldsymbol{\theta}_0) \,, \tag{2.6}$$

where $\mathbf{H}$ is the negative Hessian matrix of the log posterior distribution evaluated at

the posterior mode. Exponentiating both sides in Equation 2.6, we obtain

$$f(\boldsymbol{\theta}) \simeq f(\boldsymbol{\theta}_0) \exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^{\mathrm{T}}\mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)\right\} . \tag{2.7}$$

From (2.7), $f(\boldsymbol{\theta})$ is seen to be approximately equal to a multivariate normal density with mean $\boldsymbol{\theta}_0$ and covariance matrix $\mathbf{H}^{-1}$. Since the posterior density, $p(\boldsymbol{\theta} \,|\, \mathbf{y})$, is proportional to $f(\boldsymbol{\theta})$, it too is approximately equal to the multivariate normal density. We note this approximation assumes that the prior distribution of $\boldsymbol{\theta}$ and the likelihood must be positive and twice differentiable near the posterior mode. For further details, see [29].

To compute estimates of the parameters using the BCLT, we use the R built-in function called `nlminb`. This function implements constrained and unconstrained optimizations using PORT routines [30], allowing us to estimate the posterior mode numerically. Subsequently, we use the R function `hessian`, from the package **numDeriv** [31] to calculate a numerical approximation to the Hessian matrix of the log posterior function at the estimated posterior mode.

2.4.5 Algorithmic implementation details

B2Z is an R package that performs sampling-based Bayesian inference for the two-zone model. Currently, three sampling algorithms are available: (a) MCMC, (b) IMIS and (c) SIR. In addition, the package also offers approximate Bayesian estimation using (d) the BCLT. The Bayesian two-zone model can be fitted using the function `B2ZM`, where the desired sampling algorithm is specified as an argument to this function. Another option is to use one of the following functions directly: `B2ZM_BCLT`, `B2ZM_MCMC`, `B2ZM_IMIS` and `B2ZM_SIR`. In either cases, the output is a valid input for the functions `summary` and `plot`. For instance, suppose `fit` is an output from `B2ZM`. Then, the line command `summary(fit)` returns the following:

- Some posterior summaries for each of the parameters $\boldsymbol{\theta}$:

  - Posterior median, mean, standard deviation;
  - $100(1-\alpha)\%$ credibile intervals, where α is specified by the user;
  - Posterior covariance matrix.

- Posterior model comparisons using the deviance information criterion (DIC); see Section 3.3.2.

- Sample quality measurements that depend on the sampler algorithm. Specifically,
  - SIR: Effective sample size (ESS), proportion of unique points in the sample, maximum importance weight;
  - IMIS: ESS, maximum importance weight, variance of the re-scaled importance weights, entropy of importance weights relative to uniformity, expected fraction of unique points and expected number of unique points after re-sampling;
  - MCMC: ESS and MCMC acceptance rate.

The line command `plot(fit)` produces some graphical summaries of the estimated model. In particular, this line command returns:

- $100(1-\alpha)\%$ posterior predictive interval along with the posterior median of the log concentrations at the near field over the observed period of time, where α is specified by the user;

- $100(1-\alpha)\%$ posterior predictive interval and the posterior median of the log concentrations at the far field over the observed period of time, where α is specified by the user;

- Empirical posterior distributions for each parameter in the model;

- If Metropolis-within-Gibbs is selected, autocorrelation function (ACF) and trace history of the sampling of each parameter is also plotted.

Due to the domain of the parameters in the model, we actually implement the algorithms cited previously (except SIR) on a transformation of $\boldsymbol{\theta}$. After sampling from the posterior distribution of the transformed variables, we back transform to obtain a sample from the posterior distribution of $\boldsymbol{\theta}$. In particular, consider the dependent model and denote $x_1 = \beta$, $x_2 = Q$, $x_3 = G$, $x_4 = \tau_1$, $x_5 = \tau_2$ and $x_6 = \tau_{12}$. Suppose the priors for β, Q and G have supports (l_1, u_1), (l_2, u_2) and (l_3, u_3), respectively, where $0 \leq l_i < u_i < \infty$ for all $i = 1, 2, 3$. Consider the following transformations given by $h_i(\cdot)$ for $i = 1, 2, \ldots, 6$:

$$\begin{aligned}
\kappa_i = h_i(x_i) &= \log\left(\frac{x_i - l_i}{u_i - x_i}\right) \quad \forall \;\; i = \{1, 2, 3\}, \\
\kappa_i = h_i(x_i) &= \log\left(x_i\right) \quad \forall \;\; i = \{4, 5\}, \\
\kappa_6 = h_6(x_4, x_5, x_6) &= \log\left(\frac{x_6 + \sqrt{x_4 x_5}}{\sqrt{x_4 x_5} - x_6}\right) \; .
\end{aligned}$$

Therefore, the domain of $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_6)^{\mathrm{T}}$ is in $\mathbb{R}^6$. Notice that if the independent model is considered, the variable κ_6 is not needed in $\boldsymbol{\kappa}$, and therefore its domain is $\mathbb{R}^5$. Thus, the density of $\boldsymbol{\kappa}$ is given by:

$$p(\boldsymbol{\kappa} \,|\, \mathbf{y}) = p(\boldsymbol{\theta} = \mathbf{h}^{-1}(\boldsymbol{\kappa}) \,|\, \mathbf{y}) \cdot |\mathbf{J}|, \tag{2.8}$$

where $\mathbf{h}^{-1}(\boldsymbol{\kappa}) = (h_1^{-1}(\kappa_1), h_2^{-1}(\kappa_2), \ldots, h_6^{-1}(\kappa_6))^{\mathrm{T}}$ and $|\mathbf{J}|$ is the Jacobian of the transformation with $\mathbf{J}$ being the 6×6 matrix whose (i, j)-th element is $J_{ij} = \frac{\partial x_i}{\partial \kappa_j}$. Regardless of the model choice, the matrix $\mathbf{J}$ is a lower triangular matrix, therefore $|\mathbf{J}|$ is the

product of the diagonal elements, which is

$$|\mathbf{J}| = \prod_{i=1}^{3} \frac{(u_i - l_i)e^{\kappa_i}}{(1+e^{\kappa_i})^2} \times \prod_{i=4}^{5} e^{\kappa_i}, \qquad \text{(independent model)}$$
$$|\mathbf{J}| = \prod_{i=1}^{3} \frac{(u_i - l_i)e^{\kappa_i}}{(1+e^{\kappa_i})^2} \times \prod_{i=4}^{5} e^{\kappa_i} \times \frac{2e^{\frac{(\kappa_4+\kappa_5)}{2}+\kappa_6}}{(1+e^{\kappa_6})^2}. \qquad \text{(dependent model)} \tag{2.9}$$

The transformation $h_i(\cdot)$ makes the support of κ_i the real line, which improves algorithmic efficiency. Also, when we back transform, the sampled values for β, Q and G are within their respective domains, and the covariance matrices composed by the sampled values for τ_1, τ_2 and τ_{12} are positive definite, since h_4 and h_5 guarantee that τ_1 and τ_2 are positive, and h_6 ensures that the covariance inequality holds, that is, $\tau_{12}^2 \leq \tau_1\tau_2$.

When any of the domains of β, Q or G is $(0, \infty)$, the corresponding κ_i is the natural logarithm of x_i and the computation of the Jacobian is still very similar to the one presented in Equation 2.9.

2.5 Illustrating B2Z

In this section we illustrate **B2Z** using a synthetic dataset and a real dataset. The simulated exposure concentrations at the near and far fields, over n time points, were generated according to the following algorithm:

1. Choose the values of the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\Sigma}$ as desired. Recall that $\boldsymbol{\Sigma}$ is a diagonal matrix in the independent model, or a matrix with non-null entries in the off diagonal for the dependent model. In any case, $\boldsymbol{\Sigma}$ must be a positive definite matrix.

2. **For** (i `in` $1:n$)

(a) Using the fixed parameters in **Step 1**, find the log-solution of the system of differential equations in (2.1). Denote this solution by

$$\mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t_i) = (\log c_N(\boldsymbol{\theta}_1; \mathbf{x}, t_i), \log c_F(\boldsymbol{\theta}_1; \mathbf{x}, t_i))^{\mathrm{T}}.$$

(b) Generate the measurement error component $\boldsymbol{\epsilon}(t_i) = (\epsilon_1(t_i), \epsilon_2(t_i))^{\mathrm{T}}$ from a $\mathrm{N}_2(\mathbf{0}, \boldsymbol{\Sigma})$.

(c) The log exposure concentrations in the near and far fields at time t_i are

$$\mathbf{y}(t_i) = \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t_i) + \boldsymbol{\epsilon}(t_i).$$

(d) The exposure concentrations in the near and far fields at time t_i are $\exp(\mathbf{y}(t_i))$.

Section 2.5.1 presents an application of the Bayesian two-zone model to a simulated dataset considering dependent measurement errors, i.e., $\tau_{12} \neq 0$, while Section 2.5.2 applies the model to a real exposure dataset. We started each sampler with a seed set to 2011.

2.5.1 Simulated data

Consider a simulated dataset that contains 100 exposure concentrations equally-spaced between times 0 and 4 minutes. Following the study simulation in [7], the parameters values used in the simulation process are: $\beta = 5$ m^3/min, $Q = 13.8$ m^3/min, $G = 351.54$ mg/min and $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.64 \end{bmatrix}$. The volumes at the near and far fields in this simulated data are, respectively, $V_N = \pi \times 10^{-3}$ m^3 and $V_F = 3.8$ m^3.

To fit the BNLR, we need to specify the prior distributions for the unknown parameters. Here we use the same prior distributions as in [7]. In particular, we assume that $\beta \sim \mathrm{U}(0, 10)$, $Q \sim \mathrm{U}(11, 17)$ and $G \sim \mathrm{U}(281, 482)$. The dependent model is used in this section. Therefore, we assume that $\boldsymbol{\Sigma} \sim \mathrm{IW}\left(\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, 4\right)$.

The example code below illustrates how to specify the model information and the sampling algorithm desired using **B2Z**.

```
R> set.seed(2011)
R> fit.depend <- B2ZM(data = ex1, priorBeta = "unif(0,10)",
+                     indep.model = FALSE, priorQ = "unif(11,17)",
+                     priorG = "unif(281,482)", S = diag(10,2), v = 4,
+                     VN = pi*10^-3, VF = 3.8, sampler = "MCMC",
+                     mcmc.control = list(NUpd = 10000, burnin = 1000,
+                     lag = 1, m = 5000))
```

The argument `data` is a 3-column matrix such that the columns are time, exposure concentrations at the near field and at the far field, respectively. The argument `mcmc.control` is a list that contains the input parameters for the Metropolis-within-Gibbs algorithm. Similarly, there are control input arguments to BCLT, IMIS and SIR as well, which are `bclt.control`, `imis.control` and `sir.control`, respectively. More details about the arguments in `B2ZM` can be found in `R` using the line command `help(B2ZM)`.

As discussed in Section 2.4, the sampling algorithms require some input parameters. Table 2.1 presents the input parameters provided for each sampling algorithm in this example. The BCLT implemented in the **B2Z** package requires two input parameters: `m` and `sample_size`. In particular, to estimate the posterior mode (needed in the BCLT), the function `nlminb` is used, which depends on the starting parameter values. The input `m` is the number of sampling values from the prior distributions of β, Q and G. Therefore, the vector among the `m` sampled with largest likelihood value is used as starting parameter values. The other input parameter `sample_size` is the size of the sample from the approximate posterior distribution of the parameters in the model according to the Bayesian Central Limit Theorem. We use `m = 8000` and `sample_size = 2000`.

Table 2.2 presents several posterior summaries for each parameter in the dependent model obtained by using the function `B2ZM` within the package **B2Z**. The IMIS and Metropolis-within-Gibbs algorithms provide similar estimates for the parameters in the model. In addition, the posterior means obtained by these algorithms fairly estimate

Sampler	**Input parameters**
MCMC	$N = 10000$, $burnin = 1000$, $thin = 1$
IMIS	$N_0 = 6000$, $B = 600$, $M = 3000$
SIR	$m = 50000$

Table 2.1: Input parameters for each posterior sampling algorithm.

the parameters in the model, except for β and G estimated using the SIR algorithm. The 95% credible intervals cover the true values of the parameters, except for τ_2 when using the SIR algorithm.

Parameter	Real value	Sampler	2.5%	Median	97.5%	Mean	SD
β	5.000	SIR	3.613	7.874	7.874	6.543	1.677
		IMIS	3.728	5.091	6.859	5.158	0.801
		MCMC	3.736	5.141	6.970	5.201	0.819
		BCLT	3.562	4.985	6.351	4.963	0.716
Q	13.800	SIR	13.356	14.494	14.562	14.326	0.477
		IMIS	11.403	14.570	16.872	14.458	1.573
		MCMC	11.375	14.705	16.897	14.552	1.577
		BCLT	11.736	14.251	16.478	14.212	1.325
G	351.540	SIR	310.223	414.590	469.354	393.996	46.670
		IMIS	296.300	375.007	463.859	376.749	45.239
		MCMC	294.689	379.521	468.779	379.639	45.466
		BCLT	304.889	369.093	444.387	370.910	36.885
τ_1	1.000	SIR	0.957	0.957	1.735	1.129	0.297
		IMIS	0.984	1.283	1.738	1.302	0.192
		MCMC	0.993	1.289	1.723	1.308	0.188
		BCLT	0.963	1.263	1.662	1.275	0.179
τ_2	0.640	SIR	0.683	0.683	0.959	0.729	0.072
		IMIS	0.577	0.747	0.989	0.756	0.105
		MCMC	0.572	0.742	0.989	0.752	0.108
		BCLT	0.553	0.723	0.944	0.731	0.102
τ_{12}	0.500	SIR	0.320	0.376	0.617	0.412	0.103
		IMIS	0.375	0.565	0.826	0.576	0.117
		MCMC	0.375	0.567	0.828	0.576	0.116
		BCLT	0.359	0.561	0.792	0.565	0.111

Table 2.2: Posterior summaries - Dependent model.

In this example, the SIR algorithm samples poorly from the posterior distribution. In fact, the proportion of unique points in the sample is very low (0.062%), which explains the strange behavior in the standard deviation estimates. On the other hand, IMIS and Metropolis-within-Gibbs algorithms perform better. In particular, the IMIS has

an expected fraction of unique points equaling 58.7% and the ESS for the Metropolis-within-Gibbs the acceptance rate is 51.27%.

The following figures are produced using the line command `plot(fit.depend)`, where the output `fit.depend` is an object from the BNLR fitted using the Metropolis-within-Gibbs algorithm. The analogous figures for the SIR and IMIS algorithms and for BCLT are not shown in this chapter. However, they can be produced by running the example code in the tutorial for **B2Z**.

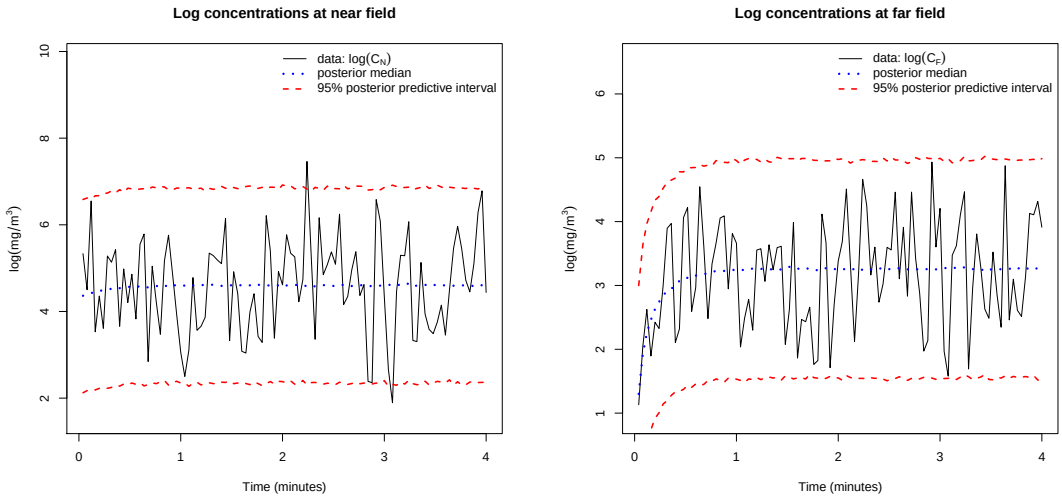


Figure 2.2: 95% posterior predictive intervals and posterior medians of the log exposure concentrations at the near and far fields over the observed period of time.

Figure 2.2 shows the 95% posterior predictive intervals and the posterior medians of the log exposure concentrations at the near and far fields. These graphs help environmental researchers know more about the range of the log exposure concentrations over the observed period of time in both fields. The solid lines in Figure 2.2 represent the observed log exposure concentrations.

Figure 2.3 shows the empirical posterior distributions of the parameters in the dependent model. Each empirical posterior distribution contains two curves:

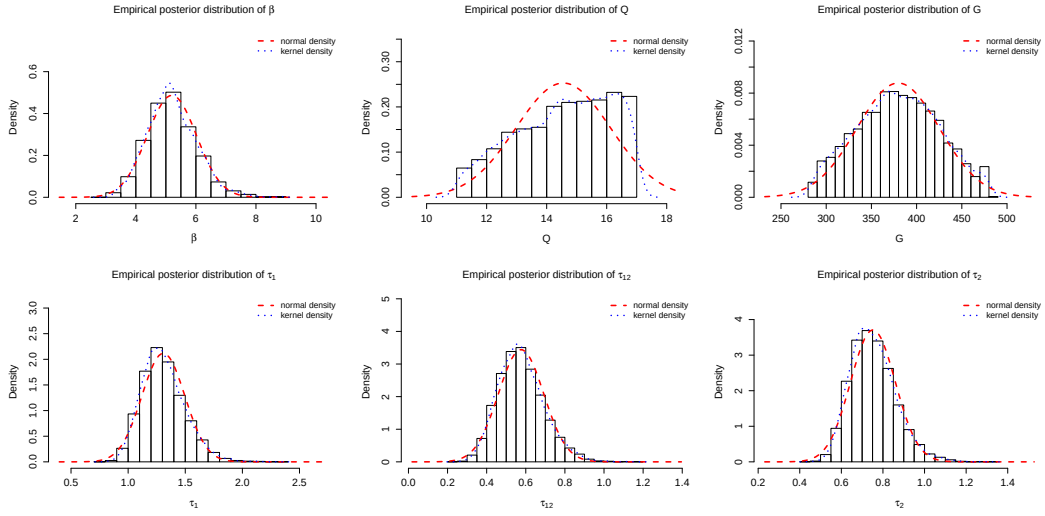


Figure 2.3: Empirical posterior distributions of the parameters in the dependent model.

- normal density centered at the estimated posterior mean and scaled by the estimated posterior standard deviation of the parameter;
- Gaussian kernel density curve.

Figure 2.4 shows the Metropolis history plot and ACF for the parameters in the model.

The BNLR fitting was done on a PC Intel Core Duo CPU P8600 with 2.40GHz and 4.00GB of Memory RAM. The computational time (in seconds) for the SIR, IMIS, Metropolis-within-Gibbs and BCLT algorithms are 59.36, 133.30, 67.40, and 18.64, respectively. In this example, the computational time for the Metropolis-within-Gibbs also includes the time spent estimating the starting values and the covariance matrix needed for the proposal distribution.

B2Z can also interact with the package **coda**. For instance, Gelman and Rubin's convergence diagnostic can be computed very easily using the function `gelman.diag` provided by the package **coda**. To compute that measure we need to fit the model more than one time. For example, suppose we fit the model three times using Metropolis

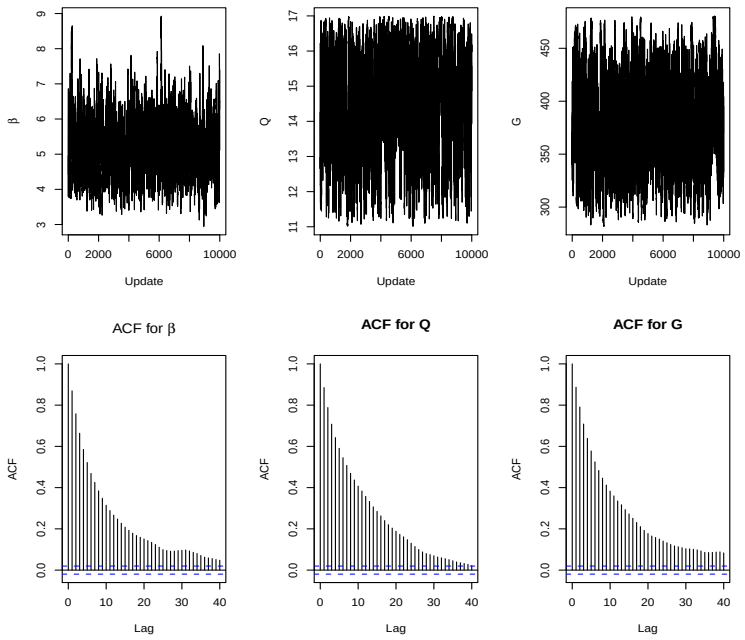


Figure 2.4: MCMC trace and ACF plots for β, Q and G.

sampler and denote them by `fit.depend1`, `fit.depend2` and `fit.depend3`. The following code shows how to compute the Gelman and Rubin's convergence diagnostic in this example.

```
R> fit1 <- do.call(cbind, fit.depend1[c("Beta","Q","G")])
R> fit2 <- do.call(cbind, fit.depend2[c("Beta","Q","G")])
R> fit3 <- do.call(cbind, fit.depend3[c("Beta","Q","G")])
R> x <- mcmc.list(list(mcmc(fit1),mcmc(fit2),mcmc(fit3)))
R> gelman.diag(x)
```

For further information regarding Gelman and Rubin's convergence diagnostic see [32]. A multivariate version of Gelman and Rubin's diagnostic was proposed by [33].

2.5.2 Experimental two-zone study

In this section we fit the Bayesian two-zone model to the data set used in the experimental two-zone study in [7]. The experiment consisted of observed exposure concentrations

of toluene over a period of time, where Q and G were known and equal to 13.8 m^3/min and 351.5 mg/min, respectively. The measurements were made in four directions (east, west, north and south) on three horizontal parallel planes at 5 different distances (10 cm, 15 cm, 20 cm, 30 cm, and 40 cm) from the contamination source, where the source was located on the middle plane and the exposure concentrations were measured every 5 seconds for at least 15 minutes in each location. Although combinations of factors such as presence of a worker's body, body movement and heat were also included in the experimental study, here we consider only the plain experimental data, i.e, the measurements that do not include any of those factors. A very detailed explanation of this experiment can be found in [7].

To illustrate **B2Z** using this real data set, we use the observed exposure concentrations on the middle plane and north direction. The measurements at 10 cm and 15 cm from the contamination source represent the exposure concentrations at the near and far fields, respectively. It is important to mention that the exposure concentrations at the near and far field were not observed simultaneously. However, we will still consider that this assumption is true since, according to [7], the exposure concentrations at the near and far fields had the same initial conditions when measured. There are 134 observed time points equally spaced between 57.08 and 68.17 minutes, and the volumes of the near and far fields are $\pi \times 10^{-3}$ m^3 and 3.8 m^3, respectively.

Following [7], we let $\beta \sim \mathrm{U}(0, 10)$, $Q \sim \mathrm{U}(11, 17)$ and $G \sim \mathrm{U}(281, 482)$. We also assume $\boldsymbol{\Sigma} \sim \mathrm{IW}\left(\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, 4\right)$. To fit this model using the IMIS sampler, we use the following line command:

```
R> fit.imis <- B2ZM(data = real.data, priorBeta = "unif(0,10)",
+                   indep.model = FALSE, priorQ ="unif(11,17)",
+                   priorG = "unif(281,482)", S = diag(10,2),
+                   v = 4, VN = pi*10^-3, VF = 3.8, sampler ="IMIS",
+                   imis.control = list( N0 = 6000, B = 600,
+                   M = 3000, it.max = 16))
```

Now we fit the BNLR using the Metropolis sampler. However, unlike the previous

section, we provide the covariance matrix in the proposal distribution for the Metropolis-within-Gibbs algorithm. To do this, we use the output `imis.control` to form a guess for such a matrix. The following line commands show how this can be done:

```
R> initial <-  summary(fit.imis)$summary[,"Mean"][1:3]
R> prop.matrix <- summary(fit.imis)$PostCovMat[1:3,1:3]
```

Therefore, defining the covariance matrix for the proposal distribution in the function `B2ZM` is very straightforward, as given in the code below:

```
R> fit.mcmc <- B2ZM(data = real.data, priorBeta = "unif(0,10)",
+               indep.model = FALSE, priorQ = "unif(11,17)",
+               priorG = "unif(281,482)", S = diag(10,2), v = 4,
+               VN = pi*10^-3, VF = 3.8, sampler = "MCMC",
+               mcmc.control = list(initial = initial,
+               Sigma.Cand = prop.matrix, NUpd = 10000, burnin = 1000,
+               lag = 1))
```

The estimates for the parameters using IMIS and Metropolis-within-Gibbs are presented in Table 2.3. This table shows that the Metropolis and IMIS algorithms yield similar estimates. More interestingly, the sampled values for Q and G are very close to the boundaries imposed by their respective prior distributions. This may indicate that the systemic component (two-zone model) and the stochastic measurement error are not enough to explain the variability in this data. In fact, observe that Figure 2.5 shows the posterior medians do not predict the log exposure concentrations very well, for both fields.

Parameter	Sampler	2.5%	Median	97.5%	Mean	SD
β	IMIS	1.727	2.092	2.618	2.111	0.222
	MCMC	1.699	2.101	2.610	2.113	0.231
Q	IMIS	11.005	11.099	11.466	11.142	0.134
	MCMC	11.003	11.081	11.487	11.124	0.130
G	IMIS	460.895	478.065	481.764	476.318	5.647
	MCMC	461.625	477.746	481.830	476.133	5.476
τ_1	IMIS	0.094	0.163	0.311	0.173	0.056
	MCMC	0.092	0.164	0.313	0.173	0.057
τ_2	IMIS	1.9696	2.425	3.232	2.469	0.309
	MCMC	1.9296	2.425	3.113	2.452	0.308
τ_{12}	IMIS	0.200	0.453	0.782	0.463	0.149
	MCMC	0.1767	0.453	0.783	0.460	0.154

Table 2.3: Posterior summaries - Experimental data set - Dependent model.

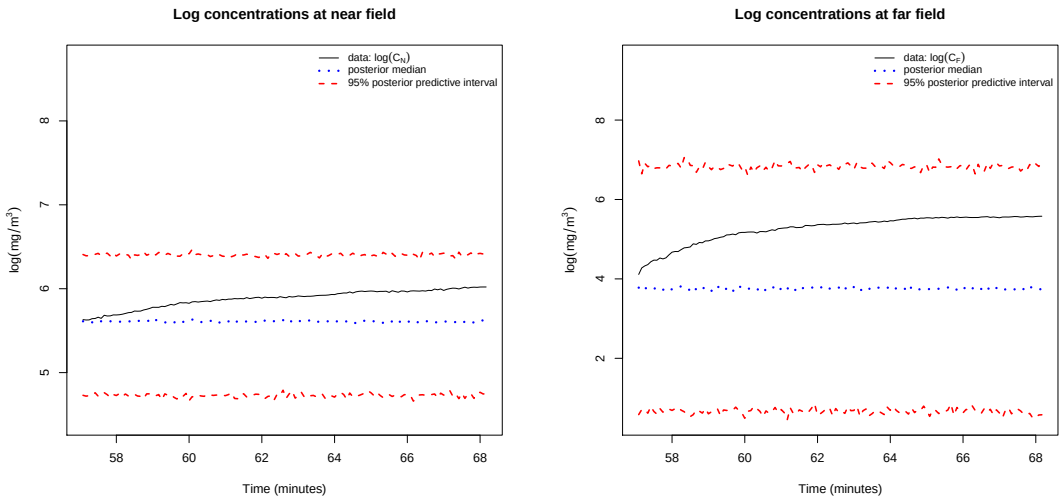


Figure 2.5: 95% posterior predictive intervals and posterior medians of the log exposure concentrations at the near and far fields over the observed period of time - Real experimental data set.

2.6 Discussion

In this chapter we have introduced our user-friendly R package **B2Z**, which is already available for download from the Comprehensive R Archive Network (CRAN) webpage (`http://CRAN.R-project.org/package=B2Z`). We have described the underlying models and a suite of algorithms to perform Bayesian inference on the two-zone models in occupational hygiene [e.g., 7]. In particular, we have demonstrated the main function called `B2ZM`, where the output from this function is a valid input for the functions `summary` and `plot`. Currently **B2Z** implements three different samplers: SIR, IMIS, and the MCMC. In addition, the package also offers approximate Bayesian inference using the Bayesian central limit theorem.

Our illustrative examples show that IMIS, MCMC and BCLT obtain similar posterior summaries. The SIR's performance was somewhat inferior, but it is easier to implement and can be useful as an initial tool for exploring approximate posteriors. Our examples also show that the BNLR fits the simulated data (which was generated from the BNLR) very well, but the performance is not as good with the real data. We believe the two components assumed in the BNLR (systematic and measurement error) are not enough to explain the all sources of variability presented in the real data. Therefore, a next step is to try a new statistical model that considers one more stochastic component which complements the two-zone model. In the next chapter, we present an approach that does this.

Chapter 3

Process-based Bayesian Melding of Occupational Exposure Models and Industrial Workplace Data

3.1 Introduction

In the previous chapter, we discovered that the BNLR predicted the exposure concentrations fairly well at the near and far fields for the simulated data, but failed to show the same effectiveness when applied to the real field data. This, to our mind, reveals that the underlying physical mechanism generating the data is too complex to be adequately captured by a simple Bayesian nonlinear regression.

To solve the above issue, we introduce in the present chapter a process-based Bayesian melding approach (PBBM) that accounts for extraneous variability in the field data that the physical models are unable to capture. This is done by adding to the BNLR a Gaussian process that complements the physical model. Although the focus in this thesis is the two-zone model, we introduce our approach using a generic physical model with multiple outcomes.

In this chapter, we also address the problem of *temporal misaligned data*, which is

common in industrial workplaces. More precisely, in the two-zone setting, we can imagine three sets of timepoints – one has observations from both the near and far-fields, another has measurements from the near-field only and the third includes measurements from the far-field only. Alternative terminology might refer to this setting as one of "missing data". In our context, such missingness is assumed *completely at random*. Interest focuses upon estimating and predicting the joint distribution of the concentrations in the two fields at any arbitrary timepoint.

The remainder of the chapter evolves as follows. Section 3.2 introduces the misaligned experimental data we subsequently analyze. Section 3.3 presents our process-based Bayesian melding approach. In Section 3.4 we compare the models' predictive performances through a simulation study and also using field data from an industrial workplace. Finally, Section 3.5 concludes the chapter by presenting some discussion with an eye toward future work.

3.2 Experimental two-zone data

In the previous chapter, the experimental data was composed by measurements that were collected at both near and far fields. However, that data in its raw format also contains measurements that were collected in one of the fields but not both. Figure 3.1 presents this two-zone experimental data set. It is divided into three "time zones". In zone I, only the near field provides measurements (83 timepoints), zone II measures both fields (134 timepoints) and, finally, zone III measures only the far field (160 timepoints) to yield $83 + 160 + 2 \times 134 = 511$ measurements.

A brief exploratory analysis of the data reveals why relying upon the physical model alone for inference and scientific deductions is undesirable. Under the assumption of zero initial concentration, and given that in this experiment $Q = 13.8$ m^3/min and $G = 351.5$ mg/min were known, the theoretical implication of the two-zone model is that the steady state concentration in the far field should be about $351.5/13.8 \approx 25$ mg/m^3.

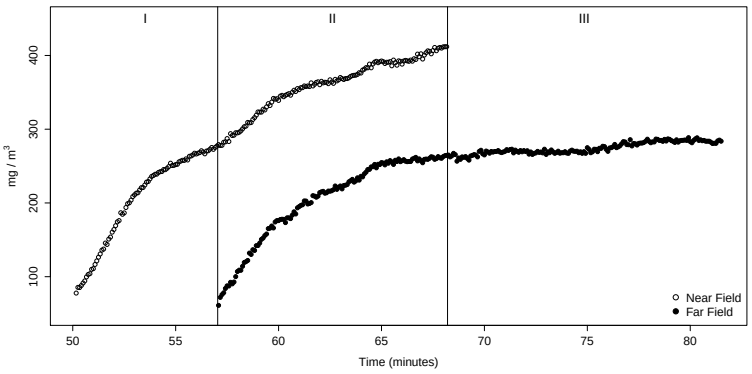


Figure 3.1: Two-zone experimental data

Consequently, methods that purely rely upon regressing on the physical model [e.g., 7] are also likely to produce grossly biased inference and poor predictions. A richer and more flexible approach is to assume an *unknown* function of time that can be estimated at arbitrary timepoints. Within a Bayesian setting, one needs a *prior* on this random function. This is achieved by a random process or a stochastic process over time. We elucidate this approach in subsequent sections.

3.3 Process-based Bayesian melding

We elucidate our approach using a generic setup that considers the following distinct modeling ingredients: (a) an $m \times 1$ vector of measurements $\mathbf{y}(t) = [y_1(t), \cdots, y_m(t)]^{\mathrm{T}}$ taken at time point t, (b) inputs (parameters), denoted as $\boldsymbol{\theta}_1$, in the physical model that are unknown, and (c) variables $\mathbf{x}$ that act as experimental controls and are known. For instance, in the two-zone model $\boldsymbol{\theta}_1 = \{\beta, Q, G\}$, $\mathbf{x} = \{V_N, V_F\}$ and $m = 2$.

Following recent research [see, e.g. 16], it is almost always beneficial to represent a physical model as a *biased* representation of "reality". This bias represents discrepancies arising due to instrumentation and/or measurement error, and the inadequacy of the physical model itself for capturing the "real" process. A stochastic mechanism to capture this bias is likely to yield better model fit and estimation of underlying variability. Thus,

we model $\mathbf{y}(t)$ as the sum of three components: (a) a systemic component represented by the physical model as a regression; (b) a stochastic process to complement the physical model; and (c) a stochastic measurement error process. We write

$$\begin{gathered} y_i(t) = f_i(\boldsymbol{\theta}_1; \mathbf{x}, t) + \eta_i(t) + \epsilon_i(t),\ i = 1, 2, \ldots, m \\ \Downarrow \\ \mathbf{y}(t) = \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t) + \boldsymbol{\eta}(t) + \boldsymbol{\epsilon}(t)\ , \end{gathered} \tag{3.1}$$

where $f_i(\boldsymbol{\theta}_1; \mathbf{x}, t)$ is the physical model, possibly transformed to a scale commensurate with $y_i(t)$, $\eta_i(t)$'s are stochastic processes that capture the bias (extraneous variability) and $\epsilon_i(t)$'s are white-noise processes capturing variation attributable to measurement error and other sources of micro-scale discrepancies. Equation (3.1) also shows the corresponding vector representation, where $\mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t)$, $\boldsymbol{\eta}(t)$ and $\boldsymbol{\epsilon}(t)$ are $m \times 1$ vectors whose i-th elements are given by $f_i(\boldsymbol{\theta}_1; \mathbf{x}, t)$, $\eta_i(t)$ and $\epsilon_i(t)$ respectively. Our approach does not require $\mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t)$ be available in closed form, but only that it be computable for any choice of inputs $\{\boldsymbol{\theta}_1, \mathbf{x}, t\}$.

3.3.1 Multivariate process models

The critical element in (3.1) is $\boldsymbol{\eta}(t)$. A natural choice is to assume an autoregressive structure for $\boldsymbol{\eta}(t)$, as is done in dynamic models. However, this treats time as discrete, which precludes estimation of the concentrations smoothly at *arbitrary* timepoints. A more flexible option is to model $\boldsymbol{\eta}(t)$ as a continous random function of time using a Gaussian process. Suppose we observe $\mathbf{y}(t)$ at n different time points $\mathbf{t} = \{t_1, \cdots, t_n\}$ and assume that $\boldsymbol{\eta}(t) \sim \mathrm{GP}_m\left(\mathbf{0}_m, \mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \cdot, \cdot)\right)$ is a zero-centered $m \times 1$ multivariate Gaussian process. Here, $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; t, t')$ is an $m \times m$ *cross-covariance* matrix function whose (i, j)-th element is the covariance between $\eta_i(t)$ and $\eta_j(t')$ for $i, j = \{1, \cdots, m\}$ and $\boldsymbol{\theta}_2$ is a collection of unknown parameters therein. The Gaussian process assumption implies that $\boldsymbol{\eta} = \left[\boldsymbol{\eta}^{\mathrm{T}}(t_1), \cdots, \boldsymbol{\eta}^{\mathrm{T}}(t_n)\right]^{\mathrm{T}}$ is distributed as an $mn \times 1$ multivariate normal distribution: $\boldsymbol{\eta} \,|\, \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t}) \sim \mathrm{N}_{mn}\left(\mathbf{0}_{mn}, \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t})\right)$, where $\Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t})$ is the $mn \times mn$ block

matrix whose (k,l)-th block is $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; t_k, t_l)$.

Clearly, care is needed when choosing $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \cdot, \cdot)$ so that $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t})$ is a symmetric and positive definite matrix. To ensure this in a flexible and computationally feasible manner, we adopt a constructive approach that assumes that $\boldsymbol{\eta}(t)$ arises as a linear transformation of a latent $p \times 1$ multivariate process whose components are independent of each other, where $1 \leq p \leq m$. This idea is adopted in the so called "linear model of coregionalization" in spatial statistics [e.g. 34] but has not, to the best of our knowledge, been used in Bayesian melding applications. To be precise, we assume $\boldsymbol{\eta}(t) = \mathbf{A}\mathbf{w}(t)$, where $\mathbf{A}$ is an $m \times p$ matrix with unknown entries, and $\mathbf{w}(t) \sim \mathrm{GP}_p\left(\mathbf{0}_p, \mathbf{C}_{\mathbf{w}}(\cdot, \cdot; \boldsymbol{\varphi})\right)$, where $\mathbf{w}(t) = [w_1(t), \cdots, w_p(t)]^{\mathrm{T}}$ is the $p \times 1$ multivariate latent process and $\boldsymbol{\varphi}$ is a collection of unknown parameters therein.

Now, assume $\mathbf{w}(t)$ has unit variance, i.e., $\mathrm{var}\{\mathbf{w}(t)\} = \mathbf{I}_p$. Accordingly, $\mathbf{C}_{\mathbf{w}}(\boldsymbol{\varphi}; t, t') = \mathrm{cov}\{\mathbf{w}(t), \mathbf{w}(t') \,|\, \boldsymbol{\varphi}\}$ is a diagonal matrix with $\rho_i(\varphi_i; t, t')$ as the i-th diagonal element, where $\rho_i(\varphi_i; t, t')$ is the correlation between $w_i(t)$ and $w_i(t')$ for all $i = \{1, \cdots, p\}$, and $\boldsymbol{\varphi} = \{\varphi_1, \cdots, \varphi_p\}$. Regardless of how close t and t' are, we assume that there is no correlation between $w_i(t)$ and $w_j(t')$, when $i \neq j$. However, recall that the random process that impacts $\mathbf{y}(t)$ is actually $\boldsymbol{\eta}(t)$, which has the cross-covariance matrix $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; t, t') = \mathrm{cov}\{\boldsymbol{\eta}(t), \boldsymbol{\eta}(t') \,|\, \boldsymbol{\theta}_2\} = \mathbf{A}\mathbf{C}_{\mathbf{w}}(\boldsymbol{\varphi}; t, t')\mathbf{A}^{\mathrm{T}}$, where $\boldsymbol{\theta}_2 = \{\mathbf{A}, \boldsymbol{\varphi}\}$. If $p < m$, then $\boldsymbol{\eta}(t)$ is a degenerate Gaussian process (i.e., the covariance matrix for any finite realization is singular) but (3.1) is still a legitimate model for $\mathbf{y}(t)$ because of the white noise $\boldsymbol{\epsilon}(t)$.

If $\mathbf{A}$ is square (i.e. $p = m$) and non-diagonal, then $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; t, t')$ will not be diagonal, which means in this case the model permits correlation between $\eta_i(t)$ and $\eta_j(t')$, even for $i \neq j$. Furthermore, when $t = t'$ we have $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; t, t) = \mathbf{A}\mathbf{A}^{\mathrm{T}}$, which means that we can, without loss of generality, set $\mathbf{A}$ to be lower-triangular to identify with the Cholesky factor of $\mathbf{C}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; t, t)$. This, however, is not strictly required and any square-root (e.g. from a spectral or singular value decomposition) will yield a valid cross-covariance matrix for $\boldsymbol{\eta}(t)$. In fact, we obtain $\boldsymbol{\eta} \,|\, \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t}) \sim \mathrm{N}_{mn}\left(\mathbf{0}_{mn}, \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t})\right)$,

where $\Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2;\mathbf{t}) = (\mathbf{I}_n \otimes \mathbf{A})\Sigma_{\mathbf{w}}(\boldsymbol{\varphi};\mathbf{t})(\mathbf{I}_n \otimes \mathbf{A}^{\mathrm{T}})$ is guaranteed to be symmetric and positive definite as long as $\mathbf{A}$ is nonsingular. Here, $\Sigma_{\mathbf{w}}(\boldsymbol{\varphi};\mathbf{t})$ denotes the covariance matrix of $\mathbf{w}$, and $\otimes$ represents the Kronecker product.

It remains, then, to choose $\rho_1(\varphi_1;\cdot,\cdot),\cdots,\rho_p(\varphi_p;\cdot,\cdot)$. These will control the smoothness of the underlying process. Had the process been an emulator for the physical model, as is often the case for complex computer models [e.g., 16], we would require the process to be smooth. A common choice is the Gaussian correlation function, $\rho_i(\varphi_i;t,t') = e^{-\varphi_i|t-t'|^2}$. We, however, use the process to model time-varying random effects representing unaccounted-for structured extraneous variation in the data. Excessive smoothness will lead to poorer fits and is not desirable. A very flexible family, controlling both smoothness and association is the Matérn correlation function

$$\rho(\phi,\nu;t,t') = \frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{|t-t'|}{\phi}\right)^{\nu}\mathcal{K}_{\nu}\left(\frac{|t-t'|}{\phi}\right);\quad \phi>0,\nu>0\,.$$

Here $\Gamma(\cdot)$ is the usual gamma function while $\mathcal{K}_{\nu}$ is a modified Bessel function of the second kind [35]. The process is $\lceil \nu-1 \rceil$ times differentiable (in the mean square sense) and ϕ determines how quickly the correlation decays over time. In particular, the correlation decays more slowly as ϕ increases [36]. We assume that $\rho_i(\varphi_i;\cdot,\cdot)$'s are Matérn functions with distinct parameters. Specifically, let $\varphi_i = \{\phi_i,\nu_i\}$ be the Matérn parameters in $\rho_i(\varphi_i;\cdot,\cdot)$, $i=1,\cdots,p$. Consequently, $\boldsymbol{\theta}_2 = \{\mathbf{A},\phi_1,\cdots,\phi_p,\nu_1,\cdots,\nu_p\}$.

Turning to the measurement error process, we assume $\boldsymbol{\epsilon}(t)\,|\,\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3) \overset{\text{iid}}{\sim} \mathrm{N}_m\left(\mathbf{0}_m,\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)\right)$ at any timepoint t. Typically $\boldsymbol{\theta}_3$ is the collection of the $m(m+1)/2$ distinct entries in $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$, where $\mathrm{var}\{\epsilon_j(t)\} = \tau_j$ denotes the j-th diagonal entry and $\mathrm{cov}\{\epsilon_i(t),\epsilon_j(t)\} = \tau_{ij}$ is the (i,j)-th entry, for $i,j=1,\ldots,m$ and $i<j$.

For a discrete set of n timepoints in $\mathbf{t}$, the $\mathbf{y}(t_i)$'s are conditionally independent and normally distributed, i.e., $\mathbf{y}(t_i)\,|\,\boldsymbol{\theta},\mathbf{w}(t_i),\mathbf{x},t_i \overset{\text{ind}}{\sim} \mathrm{N}_m\left(\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t)+\mathbf{A}\mathbf{w}(t_i),\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)\right)$,

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$. The joint posterior distribution, $p(\boldsymbol{\theta}, \mathbf{w} \,|\, \mathbf{y}, \mathbf{x}, \mathbf{t}, \boldsymbol{\gamma})$, is proportional to

$$\left[\prod_{i=1}^{n} \mathrm{N}_m\left(\mathbf{y}(t_i) \,|\, \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t) + \mathbf{A}\mathbf{w}(t_i), \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)\right)\right] \times \mathrm{N}_{pn}\left(\mathbf{w} \,|\, \mathbf{0}_{pn}, \boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\varphi}; \mathbf{t})\right) \times p(\boldsymbol{\theta} \,|\, \boldsymbol{\gamma}), \tag{3.2}$$

where $\mathrm{N}_k(\mathbf{u} \,|\, \boldsymbol{\mu}, \mathbf{S})$ denotes a k-dimensional multivariate normal density function for $\mathbf{u}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{S}$, and $\boldsymbol{\gamma}$ is the set of all hyperparameters. We suppose that the $\boldsymbol{\theta}_i$'s have independent prior distributions, i.e., $p(\boldsymbol{\theta} \,|\, \boldsymbol{\gamma}) = \prod_{i=1}^{3} p(\boldsymbol{\theta}_i \,|\, \boldsymbol{\gamma}_i)$, where $\boldsymbol{\gamma}_i$ is the set of hyperparameters related to the prior distribution of $\boldsymbol{\theta}_i$. Appendix B outlines, for the $m = 2$ case, the conditions for the identifiability of process parameters in (3.2).

Estimation of (3.2) proceeds from a Gibbs sampler with *random-walk* Metropolis steps [18]. We implement MCMC after integrating out $\mathbf{w}$ from the model to shrink the parameter space and achieve faster convergence for $\boldsymbol{\theta}$. Posterior samples of $\mathbf{w}$ can be obtained subsequently: for each posterior sample of $\boldsymbol{\theta}$, we draw a $\mathbf{w}$ from $\mathbf{w} \,|\, \boldsymbol{\theta}, \mathbf{y} \sim \mathrm{N}_{pn}(\mathbf{m}^*_{\mathbf{w}}, \boldsymbol{\Sigma}^*_{\mathbf{w}})$ with $\mathbf{m}^*_{\mathbf{w}} = \boldsymbol{\Sigma}^*_{\mathbf{w}}\left[\mathbf{I}_n \otimes (\mathbf{A}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3))(\mathbf{y} - \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, \mathbf{t}))\right]$ and $\boldsymbol{\Sigma}^*_{\mathbf{w}} = \left[\boldsymbol{\Sigma}^{-1}_{\mathbf{w}}(\boldsymbol{\varphi}, \mathbf{t}) + \mathbf{I}_n \otimes (\mathbf{A}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)\mathbf{A})\right]^{-1}$.

3.3.2 Model assessment

We will subsequently use the Deviance Information Criterion (DIC) and the Gneiting-Raftery Scoring Rule (GRS) as model comparison metrics. Let $\boldsymbol{\Omega}$ be the collection of parameters. The DIC [37] is the sum of the posterior expected deviance $\overline{D} = \mathrm{E}_{\boldsymbol{\Omega} \,|\, \mathbf{y}}\left[-2 \log p(\text{data} \,|\, \boldsymbol{\Omega})\right]$ and the effective number of parameters $p_D = \overline{D} - D(\overline{\boldsymbol{\Omega}})$, where $\overline{\boldsymbol{\Omega}}$ denotes the posterior expectation of $\boldsymbol{\Omega}$. Models with smaller DIC's are preferred. Here, we take $\boldsymbol{\Omega}$ as the collection of $\boldsymbol{\mu}_i = \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t_i)$'s and $\boldsymbol{\eta}(t_i)$'s, for $i = 1, 2 \ldots, n$, and $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$. These parameters constitute the "focus" of the DIC.

We also use Predictive Model Choice Criterion (PMCC) based upon the marginal

posterior distribution of independently *replicated data*. Let $y_j^{\text{rep}}(t_i)$ denote the replicate for $y_j(t_i)$, $\mathbf{y}^{\text{rep}}(t_i)$ be the $m \times 1$ vector with $y_j^{\text{rep}}(t_i)$ as its j-th element and $\mathbf{y}^{\text{rep}}$ be the $mn \times 1$ vector obtained by stacking up the $\mathbf{y}^{\text{rep}}(t_i)$'s. The posterior predictive distribution for $\mathbf{y}^{\text{rep}}$ is

$$\begin{aligned} p(\mathbf{y}^{\text{rep}} \,|\, \mathbf{y}) &= \int p(\mathbf{y}^{\text{rep}} \,|\, \boldsymbol{\theta}, \mathbf{w}, \mathbf{y}) p(\boldsymbol{\theta}, \mathbf{w} \,|\, \mathbf{y}) d\boldsymbol{\theta} d\mathbf{w} \\ &= \int p(\mathbf{y}^{\text{rep}} \,|\, \boldsymbol{\theta}, \mathbf{w}) p(\mathbf{w} \,|\, \boldsymbol{\theta}, \mathbf{y}) p(\boldsymbol{\theta} \,|\, \mathbf{y}) d\boldsymbol{\theta} d\mathbf{w} \,. \end{aligned} \tag{3.3}$$

Note that each $\mathbf{y}^{\text{rep}}(t_i)$ can be regarded as the "model-predicted" value for the observed $\mathbf{y}(t_i)$; see [18]. To draw samples from (3.3), we first sample $\boldsymbol{\theta}$'s from $p(\boldsymbol{\theta} \,|\, \mathbf{y})$ (using random-walk Metropolis steps). For each sampled $\boldsymbol{\theta}$, we draw $\mathbf{w}$ from $p(\mathbf{w} \,|\, \boldsymbol{\theta}, \mathbf{y})$, one-for-one given the simulated value of $\boldsymbol{\theta}$. Subsequently, we sample $\mathbf{y}^{\text{rep}}$ from $p(\mathbf{y}^{\text{rep}} \,|\, \boldsymbol{\theta}, \boldsymbol{\eta})$, one-for-one for each simulated value of $\boldsymbol{\theta}$ and $\mathbf{w}$.

[38] present a scoring rule (GRS) that depends only on the first and second moments of the predictive distribution for the *replicated data*, and penalizes departure of replicated means from the corresponding observed values (lack of fit), as well as the uncertainty in the replicated data (often reflected by over-parametrization). The GRS is

$$GRS = -\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{y_j(t_i) - \mu_{ij}^{\text{rep}}}{\sigma_{ij}^{\text{rep}}}\right)^2 - \sum_{i=1}^{n}\sum_{j=1}^{m}\log\left\{(\sigma_{ij}^{\text{rep}})^2\right\} \,,$$

where μ_{ij}^{rep} and σ_{ij}^{rep} are the mean and standard deviation respectively of $y_j^{\text{rep}}(t_i)$ in (3.3). The GRS is easily evaluated from posterior samples. Models having higher GRS are preferred.

3.3.3 Misaligned data

Section 3.3 has so far considered the ideal situation where we observe all the outcomes for every t_i. In practice, however, it is not uncommon to encounter *misaligned* or multivariate missing data in two-zone experimental settings. This means that measurements on some of the outcomes are missing at some timepoints as is the case with our data set (Section 3.2).

An advantage of our process-based framework is that inference with misaligned data can be accommodated with some minor tweaks. We elucidate with the two-zone model model ($m = 2$) setting, where $\mathbf{A}$ is 2×2, $\mathbf{w}(t) = (w_1(t), w_2(t))^{\mathrm{T}}$ and $w_1(t)$ and $w_2(t)$ are independent Gaussian processes. It helps to distinguish among three sets of timepoints. Let $\mathbf{t}_1$ be the set of timepoints that yield observations only in the near-field, $\mathbf{t}_2$ be the timepoints that yield observations only in the far-field and $\mathbf{t}_{12}$ be the timepoints yielding simultaneous measurements from both the fields. The observed data likelihood is now

$$\prod_{t \in \mathbf{t}_{12}} \mathrm{N}_2\left(\mathbf{y}(t) \,|\, \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t) + \mathbf{A}\mathbf{w}(t), \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)\right) \times \prod_{t \in \mathbf{t}_1} \mathrm{N}_1\left(y_1(t) \,|\, f_1(\boldsymbol{\theta}_1; \mathbf{x}, t) + \mathbf{a}_{1*}^{\mathrm{T}}\mathbf{w}(t), \tau_1\right) \times \prod_{t \in \mathbf{t}_2} \mathrm{N}_1\left(y_2(t) \,|\, f_2(\boldsymbol{\theta}_1; \mathbf{x}, t) + \mathbf{a}_{2*}^{\mathrm{T}}\mathbf{w}(t), \tau_2\right) \,, \tag{3.4}$$

where $\mathbf{a}_{l*}^{\mathrm{T}}$ denotes the l-th row vector of $\mathbf{A}$, for $l = 1, 2$. The joint posterior distribution can then be obtained by multiplying (3.4) by the priors as in (3.2).

For a more generic setup, some further details on implementation may be useful. Let $\mathbf{y}$ be the $nm \times 1$ vector obtained by stacking up the $\mathbf{y}(t_i)$'s. Suppose that k of its elements are observed and consequently $nm - k$ are missing. Denote by $\mathbf{y}_{\mathrm{O}}$ and $\mathbf{y}_{\mathrm{M}}$ the observed and missing data respectively. We can write $\mathbf{y}_{\mathrm{O}}$ and $\mathbf{y}_{\mathrm{M}}$ by suitably extracting elements from $\mathbf{y}$. Therefore, there are extraction matrices, $\mathbf{P}_{\mathrm{O}}$ and $\mathbf{P}_{\mathrm{M}}$, such that $\mathbf{y}_{\mathrm{O}} = \mathbf{P}_{\mathrm{O}}\mathbf{y}$ and $\mathbf{y}_{\mathrm{M}} = \mathbf{P}_{\mathrm{M}}\mathbf{y}$. The matrix $\mathbf{P}_{\mathrm{O}}$ is $k \times nm$ and $\mathbf{P}_{\mathrm{M}}$ is $(mn - k) \times mn$. Both these matrices are short and wide and have full row rank.

Bayesian inference evaluates the full posterior predictive distribution $\mathbf{y}_{\mathrm{m}}$,

$$p(\mathbf{y}_{\mathrm{m}} \,|\, \mathbf{y}_{\mathrm{o}}) = \int p(\mathbf{y}_{\mathrm{m}} \,|\, \boldsymbol{\theta}, \mathbf{y}_{\mathrm{o}}) p(\boldsymbol{\theta} \,|\, \mathbf{y}_{\mathrm{o}}) d\boldsymbol{\theta} \,. \tag{3.5}$$

Obtaining samples from (3.5) is straightforward and can be performed *after* the posterior samples of $\boldsymbol{\theta}$ have been drawn from $p(\boldsymbol{\theta} \,|\, \mathbf{y}_{\mathrm{o}})$: for each sampled $\boldsymbol{\theta}$, we draw $\mathbf{y}_{\mathrm{m}}$ from $p(\mathbf{y}_{\mathrm{m}} \,|\, \boldsymbol{\theta}, \mathbf{y}_{\mathrm{o}})$. Matters are simplified because $\mathbf{y}_{\mathrm{m}} \,|\, \boldsymbol{\theta}, \mathbf{y}_{\mathrm{o}} \sim \mathrm{N}_{nm-k}\left(\mathbf{m}(\boldsymbol{\theta}; \mathbf{t}), \mathbf{V}(\boldsymbol{\theta}; \mathbf{t})\right)$ where

$$\mathbf{m}(\boldsymbol{\theta}; \mathbf{t}) = \mathbf{P}_{\mathrm{m}} \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, \mathbf{t}) + \mathbf{P}_{\mathrm{m}} \boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \mathbf{P}_{\mathrm{o}}^{\mathrm{T}} \left(\mathbf{P}_{\mathrm{o}} \boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \mathbf{P}_{\mathrm{o}}^{\mathrm{T}}\right)^{-1} \left(\mathbf{y}_{\mathrm{o}} - \mathbf{P}_{\mathrm{o}} \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, \mathbf{t})\right);$$

$$\mathbf{V}(\boldsymbol{\theta}; \mathbf{t}) = \mathbf{P}_{\mathrm{m}} \boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \mathbf{P}_{\mathrm{m}}^{\mathrm{T}} - \mathbf{P}_{\mathrm{m}} \boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \mathbf{P}_{\mathrm{o}}^{\mathrm{T}} \left(\mathbf{P}_{\mathrm{o}} \boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \mathbf{P}_{\mathrm{o}}^{\mathrm{T}}\right)^{-1} \mathbf{P}_{\mathrm{o}} \boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \mathbf{P}_{\mathrm{m}}^{\mathrm{T}}.$$

Here, $\boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) = \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t}) + \mathbf{I}_n \otimes \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$ is the $nm \times nm$ covariance matrix for $\mathbf{y}$ given $\boldsymbol{\theta}$. Crucially, the inverses in $\mathbf{m}(\boldsymbol{\theta}; \mathbf{t})$ and $\mathbf{V}(\boldsymbol{\theta}; \mathbf{t})$ are well-defined because $\mathbf{P}_{\mathrm{o}}$ and $\mathbf{P}_{\mathrm{m}}$ have full row rank and $\boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ is nonsingular.

3.4 Data analysis

We now implement our process-based Bayesian Melding approach (PBBM) to datasets simulated from two-zone experiments as well as the experimental data described in Section 3.2. The simulation studies we perform here demonstrate the identifiability of model parameters and the flexibility and effectiveness of the PBBM approach under diverse structural specifications. In particular, we compare the performance of different association structures using three distinct specifications for $\mathbf{A}$: vector (V), diagonal (D) and lower triangular (LT). See Table 3.1. Moreover, we compare the PBBM with the simpler Bayesian non-linear regression model (BNLR), which is essentially the PBBM without the random process (i.e., $\boldsymbol{\eta}(t) = \mathbf{0}$).

Specifications for $\mathbf{A}$ depend upon the dimension of $\mathbf{w}(t)$ and the parameters in $\mathbf{C}_{\mathbf{w}}(\boldsymbol{\varphi}, \cdot, \cdot)$. For V, $\mathbf{w}(t)$ is a univariate Gaussian process ($p = 1$) and, therefore,

Table 3.1: Matrix structures for $\mathbf{A}$.

(a) V	(b) D	(c) LT
$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$	$\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$	$\begin{bmatrix} a_1 & 0 \\ a_3 & a_2 \end{bmatrix}$

$\boldsymbol{\varphi} = \{\phi_1, \nu_1\}$. For D and LT, $\mathbf{w}(t)$ is a bivariate Gaussian process ($p = 2$) and $\boldsymbol{\varphi} = \{\phi_1, \phi_2, \nu_1, \nu_2\}$. Lastly, $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3) = \begin{bmatrix} \tau_1 & \tau_{12} \\ \tau_{12} & \tau_2 \end{bmatrix}$ accommodating *correlated* measurement errors.

3.4.1 Simulation study

We first compare the performance of the models using synthetic two-zone data sets that were generated according to the PBBM and BNLR frameworks. Specifically, we simulate 100 independent data sets from the BNLR model and from each of the three PBBM specifications in Table 3.1. Each data set is composed of exposure concentrations (log-scale) in the two fields observed at 100 equally-spaced timepoints between 1 and 100 minutes. The model parameters used to generate the data are presented in Table 3.2. For all cases, $\boldsymbol{\theta}_1 = \{\beta = 7.25, Q = 15, G = 105\}$.

Table 3.2: Parameter values used to simulate the synthetic two-zone data sets.

Model	$\mathbf{A}$	Parameters									
		$\boldsymbol{\theta}_2$							$\boldsymbol{\theta}_3$		
		ϕ_1	ϕ_2	ν_1	ν_2	a_1	a_2	a_3	τ_1	τ_2	τ_{12}
PBBM	V	8	-	2.5	-	0.032	0.141	-	0.0005	0.0100	0.0020
	D	15	8	0.5	2.5	0.032	0.141	-	0.0005	0.0100	0.0020
	LT	15	8	0.5	2.5	0.032	0.062	0.127	0.0005	0.0100	0.0020
BNLR	-	-	-	-	-	-	-	-	0.0010	0.0200	0.0040

As seen in Table 3.2, the parameters associated with the marginal variance of $\mathbf{y}(t)$

(i.e., a_i's and τ_i's) were chosen to be relatively small, which is typical in actual experimental scenarios. This also offers the BNLR a fairer platform to perform effectively because larger values for the a_i's can produce more variable concentration curves that would be more congruous with models with random effects. The input parameters for the two-zone model (i.e., $\boldsymbol{\theta}_1 = \{\beta, Q, G\}$ and $\mathbf{x} = \{V_N, V_F\}$) were taken from physical considerations deemed plausible by industrial hygienists [e.g., 2]. These values simulate a workplace where the volume of the near field is equal to half of the volume of a sphere with radius 0.8m, that is, $V_N = 1.1\text{m}^3$. Moreover, we assume $V_F = 240\text{m}^3$ and zero initial concentrations in both fields. In this scenario, the theoretical steady-state concentration at the near field ($G/Q + G/\beta \approx 21.5$ mg/m^3) is roughly three times higher than that at the far field ($G/Q = 7$ mg/m^3).

Prior settings

Table 3.3 shows the prior distributions for the physical parameters. Physical considerations and expert judgment usually lead to reasonably informative priors. Indeed, the generation and ventilation rates are customarily assigned more informative priors based upon their plausible ranges. One way to determine the hyperparameters in the

Table 3.3: Prior distributions for physical parameters.

Model	β	Q	G
PBBM, BNLR	U(0, 14.5)	U(12, 18)	U(73.5, 136.5)

prior distribution for β is to write β as the product of the random airspeed (RA) at the boundary of the near field and one half of the free surface area (SA) of the near field, i.e., $\beta = \frac{1}{2}SA \times RA$. The advantage of doing this is that SA is usually available and an estimate of RA can be obtained with a non-directional anemometer, thus giving some prior information about β.

Matters are somewhat more delicate with process parameters. Unlike the physical

parameters, the process parameters cannot be gleaned from physical considerations. In particular, the ϕ_i's and ν_i's are usually weakly identifiable from the data and will require informative priors. We choose such priors based upon mechanistic considerations. For example, the ν_i's control the smoothness of the latent process. Allowing excessive smoothness for this process will not only impair inference but also cause numerical instabililities in the fitting algorithm. Therefore, we assume that $\nu_i \sim \text{DU}(9, 0.5, 2.5)$, where $X \sim \text{DU}(k, a, b)$ denotes a discrete uniform distribution such that $P(X = a + sb) = 1/k$, for $s = 0, 1, \cdots, k-1$.

The ϕ_i's control the strength of temporal correlations in the two fields. We use the *practical range* as a basis for assigning priors. The practical range is informally defined as the time separation at which the correlation has dropped close to 0, say 0.05. We assume each $\phi_i \sim \text{DU}(10, 1, 20)$, where the hyperparameters were chosen such that the prior mean for the practical range is about half of the maximum absolute time separation.

Subsequent inference is much more robust to the prior assumptions on the entries in $\mathbf{A}$. The prior distributions for the different structural specifications of $\mathbf{A}$ are shown in Table 3.4. Because the diagonal entries in $\mathbf{A}$ must be positive, so they are assigned log-normal distributions, while the off-diagonal entry is modeled with a normal distribution. Fairly vague, but proper, priors on $\mathbf{A}$ seem to render robust inference. Here $x \sim \text{LN}(\mu, \sigma)$ denotes that the natural logarithm of x is normally distributed with mean μ and standard deviation σ.

Table 3.4: Prior distributions for the unknown entries in $\mathbf{A}$

Structure	a_1	a_2	a_3
V, D	$\text{LN}(-5.7, 2.1)$	$\text{LN}(-4.3, 2.1)$	–
LT	$\text{LN}(-5.7, 2.1)$	$\text{LN}(-4.7, 2.1)$	$\text{N}(0.1, 1)$

For the simulation studies, we have 100 simulated datasets. Assigning a different

set of priors for each study is infeasible. To maintain consistency across the different generated datasets we choose hyperparameters such that the prior mean for $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ is roughly the estimated variance of $\mathbf{y}(t)$ (averaged over the 100 simulated datasets) and the prior coefficient of variation for each parameter in $\mathbf{A}$ is roughly 10.

Lastly, we adopt an inverse-Wishart (IW) distribution for $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$. Subsequent inference is robust with respect to these parameters. We assume $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3) \sim IW(\mathbf{S}, r)$, where $r = 5$ is the degrees of freedom and the inverse scale matrix is $\mathbf{S} = \begin{bmatrix} 0.002 & 0.007 \\ 0.007 & 0.038 \end{bmatrix}$. Since the expectation of $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$ equals $\dfrac{\mathbf{S}}{r-2-1}$, $\mathbf{S}$ was chosen from rough preliminary estimates of the residual variance and covariance.

Analysis of simulated datasets (not misaligned)

We divide each simulated data set into a training set and a test set. The training set consists of exposure concentrations in both fields at 70 time points randomly selected between 1 and 100 minutes. The testing set is composed of the exposure concentrations at the remaining 30 timepoints. For each model, inference was based upon $5,000$ posterior samples obtained from our MCMC algorithm after discarding the first $5,000$ iterations as burn-in. For random-walk Metropolis steps, we transformed parameters, if necessary, to have support on the real line so that normal proposals could be used and then transformed them back to the original scale.

Table 3.5 presents the DIC and the GRS, averaged over the 100 independently generated datasets, for the PBBMs and BNLR. Here, the row labels represent the model generating the data (i.e., the "true" model), while the column labels represent the model used to fit the data sets. The numbers in the parenthesis are the standard errors.

Table 3.5 reveals that, in general, both these comparison metrics suggest that "true" model (i.e. the one from which the data was generated) seems to excel. A noticeable exception occurs when the data is generated from the BNLR. In this case, the DIC score

Table 3.5: DIC and GRS metrics for non-misaligned data. The standard errors, from the 100 simulations, are shown in parenthesis.

GOF	Model	BNLR	D	LT	V
DIC	BNLR	-470.35 (14.68)	-470.38 (14.89)	-470.31 (15.11)	-470.43 (14.82)
	D	-356.83 (41.36)	-510.78 (18.44)	-507.11 (18.88)	-478.19 (25.45)
	LT	-459.62 (33.95)	-522.40 (16.39)	-535.27 (16.60)	-522.71 (18.20)
	V	-513.16 (29.60)	-546.63 (20.22)	-560.89 (17.23)	-560.09 (17.68)
GRS	BNLR	619.81 (22.10)	621.98 (22.53)	627.36 (22.84)	623.34 (22.65)
	D	606.75 (35.52)	723.08 (24.23)	717.94 (28.15)	661.50 (38.50)
	LT	597.21 (39.79)	657.23 (25.98)	703.07 (34.45)	673.43 (37.04)
	V	626.01 (48.19)	698.64 (34.52)	735.82 (27.51)	732.71 (29.44)

suggests that all models fit the data equally well, while the GRS shows that PBBM with the LT structure outperforms the others. In fact, both the DIC and GRS metrics indicate that PBBM with LT structure is always very competitive and usually excels irrespective of the underlying generating mechanism. The standard errors indicated in parenthesis shows that there seem to be no significant difference between the DIC's of the true model and the PBBM with structure LT. On the other hand, the PBBM models seem to have significantly lower DIC and GRS scores than the BNLR when the data is generated from the PBBM models. In summary, PBBM performs better than the BNLR, except when the data comes from the BNLR model, in which case they show similar performance.

We also assess models with regard to estimation of the physical parameters, $\boldsymbol{\theta}_1$. For all models the parameters in $\boldsymbol{\theta}_1$ were estimated by their posterior means. To do the comparison, we present in Table 3.6 the RMSE averaged over 100 independently generated datasets. As before, the row labels represent the data set, while the column labels represent the model used to fit the data sets, and the numbers in parenthesis are the RMSE standard errors.

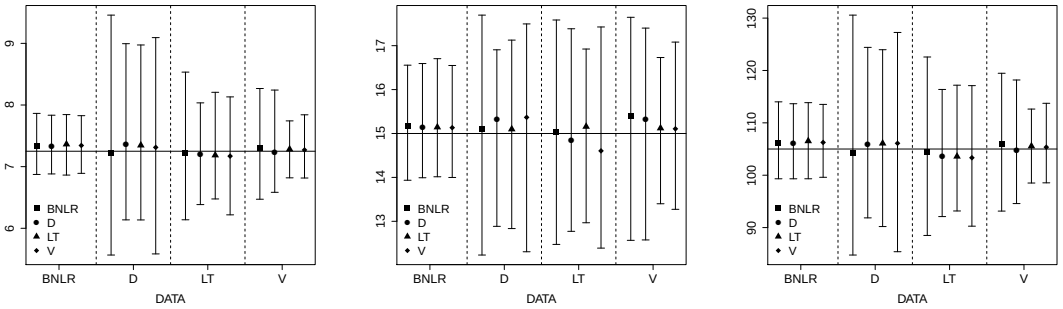
Table 3.6 reaffirms what we have seen with the model comparison metrics. Models generating the data usually perform well when fitted to that dataset. When data was generated from the BNLR models, all models estimate $\boldsymbol{\theta}_1$ similarly. We also do not find any significant differences between the RSME's from the true model and from the

Table 3.6: RMSE when estimating $\boldsymbol{\theta}_1$ - Non-misaligned data

	BNLR	D	LT	V
BNLR	3.74 (2.95)	3.80 (3.05)	3.94 (3.12)	3.75 (2.87)
D	12.27 (8.00)	8.07 (6.66)	8.18 (6.31)	11.48 (7.33)
LT	8.11 (5.92)	6.45 (4.78)	5.93 (4.48)	7.19 (5.37)
V	7.49 (4.89)	6.34 (4.27)	3.58 (2.89)	3.98 (3.32)

PBBM with LT structure.

Figures 3.2(a)-3.2(c) present summaries of the estimates that each model provided according to each dataset. The x-axis indicates the dataset, while the square, circle, triangle and diamond represent the median of the 100 independent parameter estimates according to the BNLR, D, LT and V models, respectively. Moreover, the vertical solid line represents the 2.5% and 97.5% percentiles of the parameter estimates, and the horizontal solid line depicts the true value of the parameter. These figures show that when BNLR generates the data, the BNLR and the PBBM's perform similarly. On the other hand, when PBBM's generate the data, the LT has the narrowest 95% percentile intervals of the estimates around the true value for most situations, while the BNLR has the widest 95% credible intervals of the estimates.



(a) β (b) Q (c) G

Figure 3.2: Median and 95% percentile interval of the estimates for β, Q and G - Non-misaligned data

Lastly, the story is unaltered when we compare BNLR and the PBBM's with respect to cross-validation using the hold out points. We computed the RMSE and the fraction of hold out points that lie within the 95% predictive interval (results not showed). The PBBM's present slightly smaller RMSE's than the BNLR when data are generated from PBBM's. When data are generated from BNLR, the models have essentially equal RMSE's. In terms of predictive interval coverage, all models perform very similarly.

3.4.2 Analysis of misaligned experimental data

We now analyze the misaligned experimental data described in Section 3.2. We consider the BNLR and also the PBBM approach with structures LT and D. We omit V because its performance was found to be very similar to that of D. For each model, three parallel MCMC chains were run for 30,000 iterations. Table 3.7 shows the multivariate potential scale reduction factor proposed by [33] to check MCMC convergence. For posterior analysis we discarded the first 15,000 iterations of each chain and took every 30th sample, thus obtaining a final "thinned" MCMC sample of 2,250 for each model.

Table 3.7: Multivariate Potential Scale Reduction Factor

BNLR	D	LT	GMNP
1.07	1.01	1.01	1.02

Table 3.8 presents the prior distribution adopted in each model. The • indicates the set of parameters in each model. Recall from Section 3.2 that Q and G are known for this experiment. Inferential interest focuses upon estimation of β and the subsequent estimation of the bivariate distribution for the concentrations in the two fields.

Table 3.9 shows that the PBBM significantly outperforms the BNLR. This confirms what we suspected in Section 3.2. The experimental data heavily violates the physical model assumptions. Consequently, a statistical model that only has a physical model and a measurement error components will fit the data poorly. These facts are also confirmed by Figures 3.3(a)-3.3(c), which plot the means for the replicated data against the observed log-exposure concentrations (dots). The solid line (with slope 1) represents equality between the model replicated means and the observations. Figure 3.3(a) shows

Table 3.8: Prior Distributions - Experimental Data

Parameter	Prior Distribution	Model		
		BNLR	D	LT
β	$\mathrm{U}(0, 13)$	•	•	•
a_1	$\mathrm{LN}(-4.5, 2.1)$	–	•	•
a_2	$\mathrm{LN}(-3.4, 2.1)$	–	•	–
	$\mathrm{LN}(-5.0, 2.1)$	–	–	•
a_3	$\mathrm{N}(0.3, 3)$	–	–	•
ϕ_1, ϕ_2	$\mathrm{DU}(19, 1, 5.5)$	–	•	•
ν_1, ν_2	$\mathrm{DU}(33, 0.5, 2.5)$	–	•	•
$\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$	$\mathrm{IW}\left(\begin{bmatrix} 0.0226 & 0.0715 \\ 0.0715 & 0.2359 \end{bmatrix}, 5\right)$	•	•	•

Table 3.9: DIC and GRS - Experimental data

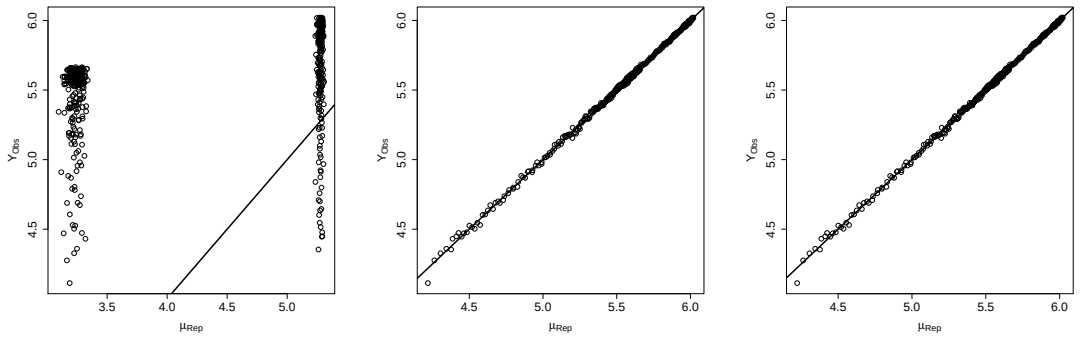
Model	DIC	p_D	$\overline{D}$	GRS
BNLR	768.56138	0.9767	767.58468	-721.9574
D	-2857.18172	36.46883	-2893.65056	3819.8092
LT	-2856.37636	37.21186	-2893.58822	3824.98638

the miserably poor fit of the simple Bayesian nonlinear regression model.

Returning to Table 3.9, we see that between the LT and D, the GRS indicates that the LT performs slightly better than the D, although the DIC seems to suggest they are quite similar. This, too, is reaffirmed by Figures 3.3(a) and 3.3(c).

Table 3.10 presents the estimated posterior means, 95% credible intervals and Monte Carlo standard errors (MCSE) (computed using non-overlapping batch means, see [39]) for the main parameters of the competing models. We see that the estimates for τ_1 and τ_2 are noticeably higher under the BNLR than under PBBMs and GMNP. This is unsurprising because the BNLR attributes the entire variation in the data to measurement errors, while PBBM attributes part of the variation to the underlying latent process as well.

Apart from these, we also see a substantial bias in β from the BNLR, which is



(a) BNLR (b) D (c) LT

Figure 3.3: Posterior replicated means versus observed log exposure concentrations - Experimental data.

Table 3.10: Posterior summaries for the main parameters in the BNLR and PBBMs

Par	Mean	95% CI	MCSE	Par	Mean	95% CI	MCSE
		BNLR				LT	
β	2.059	(1.999, 2.111)	0.004	β	6.570	(1.240, 12.587)	0.117
τ_1	0.371	(0.317, 0.434)	0.002	a_1	1.283	(0.776, 2.003)	0.017
τ_2	4.207	(3.635, 4.906)	0.012	a_2	1.391	(0.740, 2.227)	0.018
τ_{12}	1.249	(1.079, 1.453)	0.003	a_3	0.488	(-0.453, 1.576)	0.021
		D		τ_1	1.2e-04	(1e-04, 1.41e-04)	3.2e-07
β	6.437	(1.127, 12.732)	0.125	τ_2	0.001	(8.5e-04, 1.1e-03)	3.4e-06
a_1	1.245	(0.716, 1.926)	0.018	τ_{12}	2.55e-04	(2.1e-04, 3e-04)	8.4e-07
a_2	1.513	(0.977, 2.292)	0.011				
τ_1	1.2e-04	(1e-04, 1.44e-04)	4.6e-07				
τ_2	0.001	(9e-04, 1.2e-03)	4e-06				
τ_{12}	2.5e-04	(2.1e-04, 3e-04)	1.1e-06				

largely attributable to the poor fit and inadequacy of simple nonlinear regression models. What is even more disconcerting is the precise credible interval associated with β, which is likely a consequence of the BNLR's inability to adequately capture the variability. Simple least squares estimates from the observations would also suffer from such biases. The PBBM approach, on the other hand, is much more cautious and produces much wider credible intervals that seem to suggest that β cannot be estimated from the data with a lot of precision.

Lastly, we present in Figure 3.4 a panel showing the bivariate posterior predictive distributions for concentrations (in the logarithmic case) in the near and far fields as estimated by PBBM with the LT structure. In principle, our framework can produce these panels as a "movie" in continuous time. Here, we show some snapshots at nine timepoints. In each graphic, the x-axis and y-axis represent the log-concentrations in the near and far fields respectively. The first row of the panel shows density estimates at three timepoints measuring only the near field. Likewise, the second rows presents density estimates at three timepoints measuring both fields and, finally, the third row presents density estimates at three timepoints measuring only the far field. These images complete the distributional profile of the concentrations for the duration of the entire experiment and elicit the dynamic nature of the joint distribution over time.

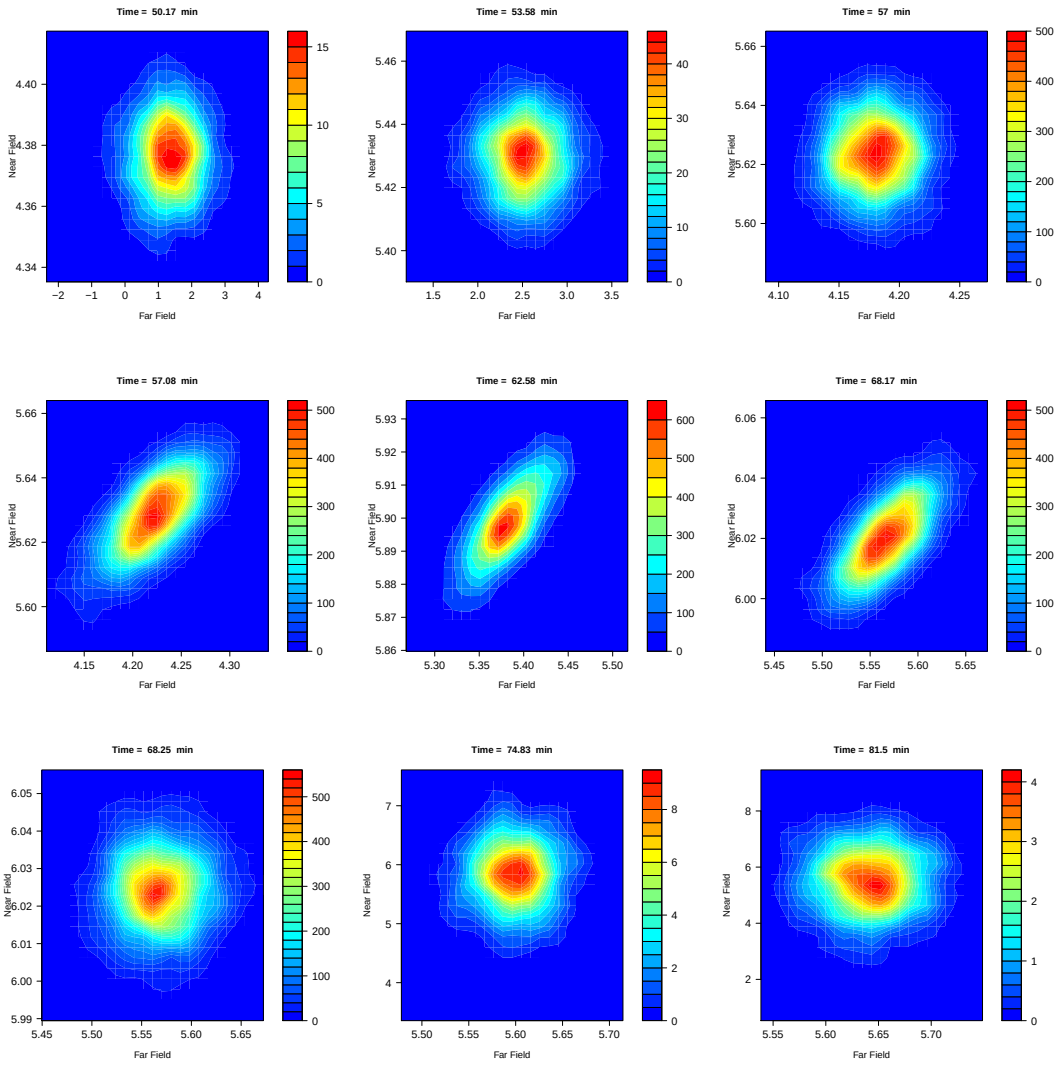


Figure 3.4: Posterior predictive joint distribution of the log-concentrations in the near and far fields at a selection of timepoints, as estimated from the PBBM with LT structure.

3.5 Discussion

In the present chapter, we introduced a process-based Bayesian melding approaches (PBBM) for predicting exposure concentrations over time in industrial workplaces. The "melding" refers to the synthesis of a posited physical model and a statistical model for the field data. The PBBM applies to workplace studies, where full inference on physical parameters is sought and subsequent predictions are carried out. Our results show that the PBBM deliver substantial, sometimes dramatic, improvements in inference than straightforward non-linear regression (BNLR).

The rich association structures in our random processes is noteworthy. Since this appears after regressing on the posited physical model, these structures can be applied even if a posited physical model were computationally prohibitive. In such cases, a distinct and smoother Gaussian process on the space of inputs can be deployed as a fast interpolator or emulator for the physical model [e.g., 16] while our specifications for $\boldsymbol{\eta}(t)$ can be used exactly as here. Also, this easily adapts to physical models with high-dimensional output [e.g., 40].

A downside of PBBM is its executing time. Fitting PBBM can take about 40 times more than fitting BNLR. Moreover, since the PBBM has more parameters, more care need to be taken in tuning the MCMC specification and checking convergence diagnostics. We believe that to attract industrial hygienists in using the PBBM we have to: (a) find a way to speed up computational time, (b) drop the "MCMC bureaucracy" and (c) still get useful results. In the next chapter we show an approximate inference for a "lighter" version of PBBM that tries to solve these problems.

Chapter 4

Fast Approximate Inference for PBBM

4.1 Introduction

In the previous chapter we showed that the PBBM approach delivered substantial improvements in inference compared to a straightforward non-linear regression. There, the PBBM inference was based on posterior samples obtained by using MCMC sampling methods such as a Gibbs sampler with random walk Metropolis steps. A disadvantage of MCMC methods is that they are usually computationally intensive and they depend upon tuning knobs and convergence diagnostics. Therefore, for an industrial hygienist unfamiliar with MCMC methods, trying to implement PBBM can be frustrating.

While we can (and will) provide a GUI software implementing the PBBM using an adaptive MCMC [e.g., 41, 42] to obviate the issues with tuning, it is unlikely that the execution time would be reduced by much. Another approach is to provide an approximate inference for the PBBM that is quickly computed and still returns useful results for the industrial hygienist. For this, we turn to recent work on analytical approximations for Gaussian process models [e.g., 43, 44, 45, 46]. In particular, we consider the integrated nested Laplace approximation (INLA) proposed by [45] because of its relative simplicity and universal applicability.

In this chapter, we employ INLA to estimate the PBBM models and compare it with MCMC. The remainder of the chapter evolves as follows. Section 4.2 briefly explains

the INLA approach and how it can be used to estimate the PBBM model. Section 4.3 analyzes INLA's performance by comparing it to a Gibbs sampling algorithm with random walk Metropolis steps. Finally, Section 4.4 concludes the chapter by presenting some discussion.

4.2 Approximate Inference

Here, we discuss an approximate inference approach for the PBBM. Our strategy is to use the integrated nested Laplace approximations (INLA) presented in [44] and [45], with some small modifications due to the PBBM's properties. We will consider the case in which (a) the ϕ_i's and ν_i's are fixed, (b) $\mathbf{A}$ is a lower triangular matrix and (c) the model has correlated error measurements, i.e.

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 \\ a_3 & a_2 \end{bmatrix} \text{ and } \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3) = \begin{bmatrix} \tau_1 & \tau_{12} \\ \tau_{12} & \tau_2 \end{bmatrix}.$$

For approximate inference, we will work on a transformation of the parameters such that each new parameter has support over the entire real line. In particular, let $x_1 = \beta$, $x_2 = Q$, $x_3 = G$, $x_4 = a_1$, $x_5 = a_2$, $x_6 = a_3$, $x_7 = \tau_1$, $x_8 = \tau_2$ and $x_9 = \tau_{12}$. Assume that the priors for β, Q and G have supports (l_1, u_1), (l_2, u_2) and (l_3, u_3), respectively, where $0 \leq l_j < u_j < \infty$ for all $j = \{1, 2, 3\}$. We then consider the following transformations given by $h_i(\cdot)$ for $i = \{1, 2, \ldots, 9\}$:

$$\begin{aligned}
\kappa_i &= h_i(x_i) = \log\left(\frac{x_i - l_i}{u_i - x_i}\right) \quad \forall \;\; i = \{1, 2, 3\}, \\
\kappa_i &= h_i(x_i) = \log(x_i) \quad \forall \;\; i = \{4, 5, 7, 8\}, \\
\kappa_6 &= h_6(x_6) = x_6 \\
\kappa_9 &= h_9(x_7, x_8, x_9) = \log\left(\frac{x_9 + \sqrt{x_7 x_8}}{\sqrt{x_7 x_8} - x_9}\right).
\end{aligned}$$

Consequently, the domain of $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_9)^{\mathrm{T}}$ is in $\mathbb{R}^9$ and

$$p(\boldsymbol{\kappa} = \boldsymbol{\kappa} \,|\, \mathbf{y}) = p(\boldsymbol{\theta} = \mathbf{h}^{-1}(\boldsymbol{\kappa}) \,|\, \mathbf{y})|\mathbf{J}|$$

where $\mathbf{h}^{-1}(\boldsymbol{\kappa}) = (h_1^{-1}(\kappa_1), h_2^{-1}(\kappa_2), \ldots, h_9^{-1}(\kappa_9))^{\mathrm{T}}$ and $|\mathbf{J}|$ is the Jacobian of the transformation with $\mathbf{J}$ being the 9×9 matrix whose (m,n)-th element is $J_{m,n} = \frac{\partial x_m}{\partial \kappa_n}$. The matrix $\mathbf{J}$ is lower triangular, therefore $|\mathbf{J}|$ is the product of the diagonal elements as below:

$$|\mathbf{J}| = \prod_{i=1}^{3} \frac{(u_i - l_i)e^{\kappa_i}}{(1+e^{\kappa_i})^2} \times \prod_{\substack{i=4 \\ i \neq 6}}^{8} e^{\kappa_i} \times \frac{2e^{\frac{(\kappa_7+\kappa_8)}{2}+\kappa_9}}{(1+e^{\kappa_9})^2}.$$

The posterior marginal distributions for the $\eta_i(t_j)$'s and κ_v's are:

$$p\left(\eta_i(t_j) \,|\, \mathbf{y}\right) = \int p\left(\eta_i(t_j) \,|\, \boldsymbol{\kappa}, \mathbf{y}\right) p\left(\boldsymbol{\kappa} \,|\, \mathbf{y}\right) \partial\boldsymbol{\kappa}$$
$$p\left(\kappa_v \,|\, \mathbf{y}\right) = \int p\left(\boldsymbol{\kappa} \,|\, \mathbf{y}\right) \partial\boldsymbol{\kappa}_{-v}$$

for $v = \{1, 2, \ldots, 9\}$, $i = \{1, 2\}$ and $j = \{1, 2, \ldots, n\}$. However, such integrals are not analytically tractable and, therefore, are computed numerically. [45] approximate these integrals using Laplace approximations for $p\left(\eta_i(t_j) \,|\, \boldsymbol{\kappa}, \mathbf{y}\right)$'s and for $p(\boldsymbol{\kappa} \,|\, \mathbf{y})$. Since in our approach $\eta_i(t_j) \,|\, \boldsymbol{\kappa}, \mathbf{y}$ is naturally Gaussian, this step is unnecessary. In fact, $\eta_i(t_j) \,|\, \boldsymbol{\kappa}, \mathbf{y} \sim \mathrm{N}_1\left(\mu^*_{\eta_{ij}}, \Sigma^*_{\eta_{ij,ij}}\right)$, where $\Sigma^*_{\eta_{ij,ij}}$ and $\mu^*_{\eta_{ij}}$ are the (ij,ij)-th and (ij)-th elements of $\boldsymbol{\Sigma}^*_{\boldsymbol{\eta}} = \left[\mathbf{I}_n \otimes \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3) + \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_2; \mathbf{t})\right]^{-1}$ and $\boldsymbol{\mu}^*_{\boldsymbol{\eta}} = \boldsymbol{\Sigma}^*_{\boldsymbol{\eta}}\left[\mathbf{I}_n \otimes \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)\right]\left[\mathbf{y} - \mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, \mathbf{t})\right]$, respectively.

Following [45], we compute the approximation for the marginal distribution of $\eta_i(t_j) \,|\, \mathbf{y}$ as follows

$$p\left(\eta_i(t_j) \,|\, \mathbf{y}\right) \approx \tilde{p}\left(\eta_i(t_j) \,|\, \mathbf{y}\right) = \sum_{k=1}^{K} p\left(\eta_i(t_j) \,|\, \boldsymbol{\kappa} = \boldsymbol{\kappa}^{(k)}, \mathbf{y}\right) p\left(\boldsymbol{\kappa} = \boldsymbol{\kappa}^{(k)} \,|\, \mathbf{y}\right) \Delta_k \,, \qquad (4.1)$$

where $\boldsymbol{\kappa}^{(k)}$ belongs to a selected set of configurations $\Gamma = \left\{\boldsymbol{\kappa}^{(1)}, \boldsymbol{\kappa}^{(2)}, \ldots, \boldsymbol{\kappa}^{(K)}\right\}$ from the parameter space and Δ_k's are the integration weights. To compute $\tilde{p}\left(\eta_i(t_j) \,|\, \mathbf{y}\right)$ we must first select the set of points Γ. [45] present two ways of doing such a task: (a) grid strategy and (b) central composite design strategy. The grid strategy constructs a d dimensional grid of points to explore $p(\boldsymbol{\kappa} \,|\, \mathbf{y})$ and locates the bulk of the probability mass. Here, d denotes the number of parameters. In our case, $d = 9$. Although intuitive and easily implemented, this method is only feasible for a scenario where the number of parameters in the model is small (less than 6). For a moderate number of

parameters (between 6 and 12), [45] suggest considering the integration problem as a response surface problem by using the central composite design (CCD) [47]. We will briefly describe this strategy in the next section.

4.2.1 Exploring $p(\boldsymbol{\kappa} \mid \mathbf{y})$ using CCD

As mentioned in the previous section, we need to choose points in the parameter space for carrying out the numerical integration in (4.1). This raises the question of how to do such a task. Obviously, we cannot simply select these points randomly nor spend too much time searching for the "perfect" set of points since that would be defeat the purpose of speeding up computation. [45] present a clever and efficient solution to this problem. The idea is to find the posterior mode of the parameters and use the CCD to find points that allow us to learn about the curvature of $p(\boldsymbol{\kappa} \mid \mathbf{y})$ around the mode. The CCD is the most popular design available for fitting a second-order response surface model. It comprises n_d factorial points from a fractional factorial design, axial points and a central point. The factorial points and the axial points lie on the surface of a d dimensional sphere with radius $\sqrt{d}$ times an arbitrary scaling σ_{ccd}.

[48] explain how to compute the locations of the fractional factorial design points. The axial points are located along each axis at distance $\pm\sigma_{ccd}\sqrt{d}$, therefore there are always $2d$ axial points. We present in Table 4.1 the total number of points $K = n_d + 2d + 1$, for some values of d.

Table 4.1: Total number of selected points according to d

d	6	7	8	9	10	11	12
K	45	79	81	147	149	151	281

[45] suggest selecting the points in the CCD strategy using a standardized $\mathbf{z}$ parametrization, defined as follows:

$$\boldsymbol{\kappa} = g(\mathbf{z}) = \boldsymbol{\kappa}^* + \mathbf{M}\mathbf{z}$$

where $\boldsymbol{\kappa}^*$ is the estimated posterior mode, and $\mathbf{M}\mathbf{M}^{\mathrm{T}}$ is the inverse of the negative Hessian at the modal configuration, which we will denote by $\boldsymbol{\Sigma}_{\boldsymbol{\kappa}}$. Originally, [45] compute

$\mathbf{M}$ based on the eigen-decomposition of $\boldsymbol{\Sigma}_{\boldsymbol{\kappa}}$. However, we will use the Cholesky decomposition, since it is more useful for our model when computing the marginal distribution of the parameters.

In order to capture the asymmetry present in $p\left(\boldsymbol{\kappa} \,|\, \mathbf{y}\right)$, [44] suggest allowing the scaling parameter σ_{ccd} to vary, not only according to the d different axis but also according to the direction, positive or negative, of each axes in the $\mathbf{z}$ parametrization. This means, we have $2d$ scaling parameters, $\left(\sigma_{ccd}^{l-}, \sigma_{ccd}^{l+}\right)$, $l = \{1, 2, \ldots, d\}$. These scaling parameters are computed such that the drop in $\log\left[p(\boldsymbol{\kappa} \,|\, \mathbf{y})\right]$ when we move from the mode to ± 2 the standard deviation is approximately 2.

Therefore, using the CCD we select K points $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(K)}$ and compute (4.1) as:

$$\tilde{p}\left(\eta_i(t_j) \,|\, \mathbf{y}\right) = \sum_{k=1}^{K} p\left(\eta_i(t_j) \,|\, \boldsymbol{\kappa} = g\left(\boldsymbol{\sigma}_{ccd}^{(k)} \circ \mathbf{z}^{(k)}\right), \mathbf{y}\right) p\left(\boldsymbol{\kappa} = g\left(\boldsymbol{\sigma}_{ccd}^{(k)} \circ \mathbf{z}^{(k)}\right) \,|\, \mathbf{y}\right) \Delta_k,$$

where $\circ$ denotes the Hadamard product and the l-th element of $\boldsymbol{\sigma}_{ccd}^{(k)}$ is defined as

$$\sigma_{ccdl}^{(k)} = \begin{cases} \sigma_{ccd}^{l-} & \text{if} \quad z_l^{(k)} < 0 \\ \sigma_{ccd}^{l+} & \text{if} \quad z_l^{(k)} > 0 \end{cases} \quad \forall \;\; l = 1, 2, \ldots, 9.$$

The integration weights for the points on the sphere with radius $f_0\sqrt{d}$ are given by

$$\Delta = \left[(K-1)\exp\left(-\frac{df_0^2}{2}\right)\left(f_0^2 - 1\right)\right]^{-1}$$

where $f_0 > 1$ is any constant. The integration weight for the central point (i.e, $\mathbf{z} = \mathbf{0}$) is 1. [46] presents details about the derivation of the weights.

4.2.2 Marginal distribution of κ_v

To compute the marginal distribution of κ_v, we integrate out the remaining parameters $\boldsymbol{\kappa}_{-v}$ from $p(\boldsymbol{\kappa} \,|\, \mathbf{y})$, i.e

$$p(\kappa_v \,|\, \mathbf{y}) = \int p(\boldsymbol{\kappa} \,|\, \mathbf{y}) d\boldsymbol{\kappa}_{-v} \;\; \forall \;\; v = \{1, 2, \ldots, 9\}.$$

However, numerically computing the marginal distribution for κ_v directly from $p(\boldsymbol{\kappa} \,|\, \mathbf{y})$ can be expensive.

One way to compute the marginal distribution of κ_v is to assume a Gaussian approximation to $\boldsymbol{\kappa} \,|\, \mathbf{y}$, i.e. $\boldsymbol{\kappa} \,|\, \mathbf{y} \approx \mathrm{N}_9\left(\boldsymbol{\kappa}^*, \boldsymbol{\Sigma}_{\boldsymbol{\kappa}}\right)$, where $\boldsymbol{\kappa}^*$ is the posterior mode and $\boldsymbol{\Sigma}_{\boldsymbol{\kappa}}$ is the inverse of the negative Hessian evaluated at $\boldsymbol{\kappa}^*$. Consequently, $\kappa_v \,|\, \mathbf{y} \approx \mathrm{N}_1\left(\kappa_v^*, \Sigma_{\kappa v,v}\right)$. This approach is known as the Bayesian Central Limit Theorem, and we will refer it as BCLT.

Another approach is to approximate $p(\boldsymbol{\kappa} \,|\, \mathbf{y})$ by a function $f(\boldsymbol{\kappa})$ that can be quickly computed. In particular, [44] suggests approximating $p(\boldsymbol{\kappa} \,|\, \mathbf{y})$ by a function $f(\boldsymbol{\kappa})$ that is based on the scaling parameters $\boldsymbol{\sigma}_{ccd}$ as follows:

$$f(\boldsymbol{\kappa}) = \prod_{m=1}^{9} f_m\left(\frac{g^{-1}(\boldsymbol{\kappa})}{\sigma_{ccdm}}\right) \tag{4.2}$$

where $f_m(\mathbf{x}) = \exp\left(-\frac{1}{2}\left(\mathbf{e}_m^{\mathrm{T}}\mathbf{x}\right)^2\right)$ and $\mathbf{e}_m$ is a canonical basis for a Euclidean space, that is, the m-th entry of $\mathbf{e}_m$ is 1 and all the remaining entries are zeros. We will refer to this approach as APFF (Approximate Posterior by a Fast Function).

Now, consider the blocks of the parameters $\mathbf{b}_p = \{\kappa_{3p-2}, \kappa_{3p-1}, \kappa_{3p}\},\ p = \{1, 2, 3\}$. Based upon our experience, the correlations within each block are high and between the blocks are considerably small. Thus, if we "force" $\mathrm{cov}(\mathbf{b}_i, \mathbf{b}_j) = \mathbf{0}_3$, $\boldsymbol{\Sigma}_{\boldsymbol{\kappa}}$ is a block matrix. Consequently, using the Cholesky decomposition, $\mathbf{M}^{-1}$ is also a block matrix. Therefore, to compute an approximation for $p(\kappa_v \,|\, \mathbf{y})$, we only need to integrate out $f(\boldsymbol{\kappa})$ with respect to the remaining two parameters that belong to the same block as κ_v. For instance,

$$p(\kappa_1 \,|\, \mathbf{y}) \approx \tilde{p}(\kappa_1 \,|\, \mathbf{y}) = \int\int f(\boldsymbol{\kappa}) \partial\kappa_2 \partial\kappa_3.$$

Once we obtain the posterior marginals for $\boldsymbol{\kappa}_v$'s we then compute the marginals for x_v's by computing the appropriate Jacobian and completing the back transformation operation. Only for τ_{12} we need to complete more steps. In particular, we find the joint distribution of $\tau_{12}, V_1 = \tau_1/\tau_2$, and $V_2 = \tau_1\tau_2$. Then, we integrate this joint distribution with respect to V_1 and V_2.

4.2.3 Computing goodness-of-fit measurements

Goodness-of-fit measurements such as DIC and the GRS are easily implemented. In particular, let $D_j\left[\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t_j),\boldsymbol{\eta}(t_j),\boldsymbol{\theta}_3\right] = -2\log\left[p(\mathbf{y}(t_j)\,|\,\boldsymbol{\theta}_1,\boldsymbol{\eta}(t_j),\boldsymbol{\theta}_3)\right]$. Then, the deviance function is

$$D(\mathbf{f},\boldsymbol{\eta},\boldsymbol{\theta}_3) = \sum_{j=1}^{n} D_j\left[\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t_j),\boldsymbol{\eta}(t_j),\boldsymbol{\theta}_3\right].$$

Therefore, the DIC is calculated as the sum of $\bar{D} = \sum_{j=1}^{n} E\left\{D_j\left[\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t),\boldsymbol{\eta}(t),\boldsymbol{\theta}_3\right]\right\}$ (posterior expected deviance) and $p_D = \bar{D} - \sum_{j=1}^{n} D_j\left(\bar{\mathbf{f}}_j,\bar{\boldsymbol{\eta}}_j,\bar{\boldsymbol{\theta}}_3\right)$ (effective number of parameters), where $\bar{\mathbf{f}}_j$, $\bar{\boldsymbol{\eta}}_j$ and $\bar{\boldsymbol{\theta}}_3$ denote the posterior expectations of $\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t_j)$, $\boldsymbol{\eta}(t_j)$ and $\boldsymbol{\theta}_3$, respectively.

We approximate $\mathrm{E}\left\{D_j\left[\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t),\boldsymbol{\eta}(t),\boldsymbol{\theta}_3\right]\right\}$ as follows:

$$\begin{aligned}\mathrm{E}\left\{D_j\left[\mathbf{f}(\boldsymbol{\theta}_1;\mathbf{x},t),\boldsymbol{\eta}(t),\boldsymbol{\theta}_3\right]\right\} \approx -2\sum_{k=1}^{K}\int \log\left[p\left(\mathbf{y}(t_j)\,|\,\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(k)},\boldsymbol{\eta}(t_j),\boldsymbol{\theta}_3=\boldsymbol{\theta}_3^{(k)}\right)\right] \times \\ p\left(\boldsymbol{\eta}(t_j)\,|\,\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(k)},\boldsymbol{\theta}_3=\boldsymbol{\theta}_3^{(k)},\mathbf{y}\right)\partial\boldsymbol{\eta}(t_j) \times \\ p\left(\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(k)},\boldsymbol{\theta}_3=\boldsymbol{\theta}_3^{(k)}\,|\,\mathbf{y}\right)\Delta_k,\end{aligned} \tag{4.3}$$

where $\boldsymbol{\theta}_1^{(k)}$ and $\boldsymbol{\theta}_3^{(k)}$ are the corresponding back-transformations of the selected point $\boldsymbol{\kappa}^{(k)}$. Notice that the bivariate integral in (4.3) can be solved analytically using the properties of the normal distribution.

Similarly, we computed an approximation for the distribution of the replicated data $y_i^{\mathrm{rep}}(t_j)\,|\,\mathbf{y}$:

$$\begin{aligned}p\left(y_i^{\mathrm{rep}}(t_j)\,|\,\mathbf{y}\right) \approx \sum_{k=1}^{K}\int p\left(y_i^{\mathrm{rep}}(t_j)\,|\,\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(k)},\eta_i(t_j),\boldsymbol{\theta}_3=\boldsymbol{\theta}_3^{(k)}\right) \times \\ p\left(\eta_i(t_j)\,|\,\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(k)},\boldsymbol{\theta}_3=\boldsymbol{\theta}_3^{(k)},\mathbf{y}\right)\partial\eta_i(t_j) \times \\ p\left(\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(k)},\boldsymbol{\theta}_3=\boldsymbol{\theta}_3^{(k)}\,|\,\mathbf{y}\right)\Delta_k\end{aligned} \tag{4.4}$$

The integral in (4.4) is also computed analytically. After obtaining the approximation

for $y_i^{\text{rep}}(t_j) \,|\, \mathbf{y}$, we numerically compute $\mu_{ij}^{\text{rep}} = \mathrm{E}\left[y_i^{\text{rep}}(t_j) \,|\, \mathbf{y}\right]$ and $\left(\sigma_{ij}^{\text{rep}}\right)^2 = Var\left[y_i^{\text{rep}}(t_j) \,|\, \mathbf{y}\right]$ for all $j = \{1, \ldots, n\}$ and $i = \{1, 2\}$. Thus, we use the μ_{ij}^{rep}'s and the σ_{ij}^{rep}'s to compute the GRS, which is given by:

$$GRS = -\sum_{i=1}^{2}\sum_{j=1}^{n}\left(\frac{y_j(t_i) - \mu_{ij}^{\text{rep}}}{\sigma_{ij}^{\text{rep}}}\right)^2 - \sum_{i=1}^{2}\sum_{j=1}^{n}\log\left\{(\sigma_{ij}^{\text{rep}})^2\right\}.$$

4.2.4 Algorithm and implementation

Below, we present the algorithm to fit the PBBM using INLA.

Algorithm 1 PBBM approximate inference algorithm

Estimate posterior mode $\boldsymbol{\kappa}^*$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\kappa}}$
Compute scaling parameters $\left(\sigma_{ccd}^{l-}, \sigma_{ccd}^{l+}\right)$, $l = \{1, 2, \ldots, d\}$
for v `in` $1:9$ **do**
  Compute the marginal posterior of κ_v
end for
for j `in` $1:n$ **do**
  Compute $E\left\{D_j\left[\mathbf{f}(\boldsymbol{\theta}_1; \mathbf{x}, t), \boldsymbol{\eta}(t), \boldsymbol{\theta}_3\right]\right\}$
  for i `in` $1:2$ **do**
    Compute $\tilde{p}\left(\eta_i(t_j) \,|\, \mathbf{y}\right)$
    Compute the approximation for $y_i^{\text{rep}}(t_j) \,|\, \mathbf{y}$
  end for
end for
Compute DIC and GRS

One great advantage of the INLA approach is that it benefits enormously from parallel programming. In Algorithm 1, notice that it takes almost no effort to separate the algorithms steps into parallel tasks (even the steps within the loops can be solved independently), i.e., fitting the PBBM using INLA is what computational scientists call an *embarrassingly parallel problem*. We implemented Algorithm 1 in R code. Table 4.2 shows the main packages we used to successfully implement it.

Table 4.2: R packages used to implement PBBM using INLA

Package	Function	Reason
multicore	mclapply	Parallel processing
R2Cuba	cuhre	Multidimensional Numerical Integration
nlme	fdHess	Evaluate an approximate Hessian Matrix
stats	nlm	Compute posterior mode

4.3 Comparison Studies

In this section we evaluate the performance of the INLA approach applied to the PBBM. More specifically, we compare the PBBM fit using INLA to the PPBM fit using the conventional Metropolis-within-Gibbs (which we will denote as MCMC). First, we make this comparison considering synthetic data (Section 4.3.1), and then considering experimental data (Section 4.3.2). In both cases, the MCMC inference was based upon 1,500 posterior samples obtained after discarding the first 5,000 iterations as burn-in and keeping every 30th draw of 3 parallel chains (each of length 20,000).

4.3.1 Simulated Data

We first examine INLA's perfomance using a synthetic two-zone data set that was generated according to the PBBM framework. Specifically, the data set is composed of exposure concentrations (log-scale) in the two fields observed at 100 equally-spaced timepoints between 1 and 100 minutes. We present the parameters used to generate this data set in Table 4.3. As mentioned before, in doing the analysis, we will consider

Table 4.3: Parameter values used to simulate the synthetic two-zone data set.

Parameters												
β	Q	G	ϕ_1	ϕ_2	ν_1	ν_2	a_1	a_2	a_3	τ_1	τ_2	τ_{12}
7.25	15	105	15	8	0.5	2.5	0.032	0.062	0.127	0.0005	0.0100	0.0020

that the ϕ_i's and ν_i's are fixed. In particular, we choose values for them such that the practical range is about half of the maximum time separation (i.e., approximately 49.5). More specifically, $\phi_1 = \phi_2 = 10.5$ and $\nu_1 = \nu_2 = 1.5$. In Table 4.4, we present the

adopted prior distributions.

Table 4.4: Prior Distributions - Synthetic Data

Parameter	Prior Distribution
β	$\mathrm{U}(0, 14.5)$
Q	$\mathrm{U}(12, 18)$
G	$\mathrm{U}(73.5, 136.5)$
a_1	$\mathrm{LN}(-5.7, 2.1)$
a_2	$\mathrm{LN}(-4.7, 2.1)$
a_3	$\mathrm{N}(0.1, 1)$
$\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$	$\mathrm{IW}\left(\begin{bmatrix} 0.0023 & 0.0069 \\ 0.0069 & 0.0381 \end{bmatrix}, 5\right)$

Table 4.5 shows the potential scale reduction factor for each parameter, computed using 3 parallel MCMC chains with different starting points. Since the 97.5% quantiles are close to 1, we conclude that there is no sign of non-convergence of the MCMC chain.

Table 4.5: Potential scale reduction factors - Synthetic Data

Parameter	Point est.	97.5% quantile
β	1.02	1.05
Q	1.00	1.01
G	1.02	1.06
a_1	1.00	1.00
a_2	1.00	1.01
a_3	1.00	1.00
τ_1	1.00	1.01
τ_2	1.00	1.00
τ_{12}	1.00	1.00

We start the comparison by first checking the estimates of the PBBM parameters. Table 4.6 presents, for each parameter in the model, the estimated posterior mean and the 95% credible interval computed using MCMC and the two approaches presented in Section 4.2.2: BCLT and APPF. In addition, Table 4.6 presents the MCSE in the square brackets for the estimates using MCMC.

Table 4.6: Posterior summaries for parameters in the PBBM - Synthetic Data

$\boldsymbol{\theta}$	MCMC	BCLT	APFF
β	7.6 (6.6, 8.9) [0.02]	7.5 (6.6, 8.5)	7.5 (6.6, 8.4)
Q	14.8 (12.2, 17.7) [0.06]	14.5 (12.4, 17.1)	14.6 (12.6, 16.9)
G	110.1 (95.4, 129.3) [0.37]	108.8 (95.6, 121.2)	108.4 (95.2, 120.8)
a_1	0.05 (0.019, 0.080) [0.001]	0.041 (0.023, 0.068)	0.042 (0.021, 0.076)
a_2	0.076 (0.032, 0.128) [0.001]	0.071 (0.042, 0.113)	0.071 (0.0412, 0.112)
a_3	0.132 (-0.001, 0.295) [0.003]	0.114 (-0.005, 0.221)	0.118 (-0.006, 0.241)
τ_1	0.0006 (0.0004, 0.0009) [3.1e-06]	0.0006 (0.0005, 0.0008)	0.0006 (0.0004, 0.0008)
τ_2	0.0123 (0.0090, 0.0169) [6.1e-05]	0.0121 (0.0089, 0.0162)	0.0120 (0.0086, 0.0163)
τ_{12}	0.0025 (0.0018, 0.0035)[1.3e-05]	0.0025 (0.0018, 0.0033)	0.0024 (0.0017, 0.0034)

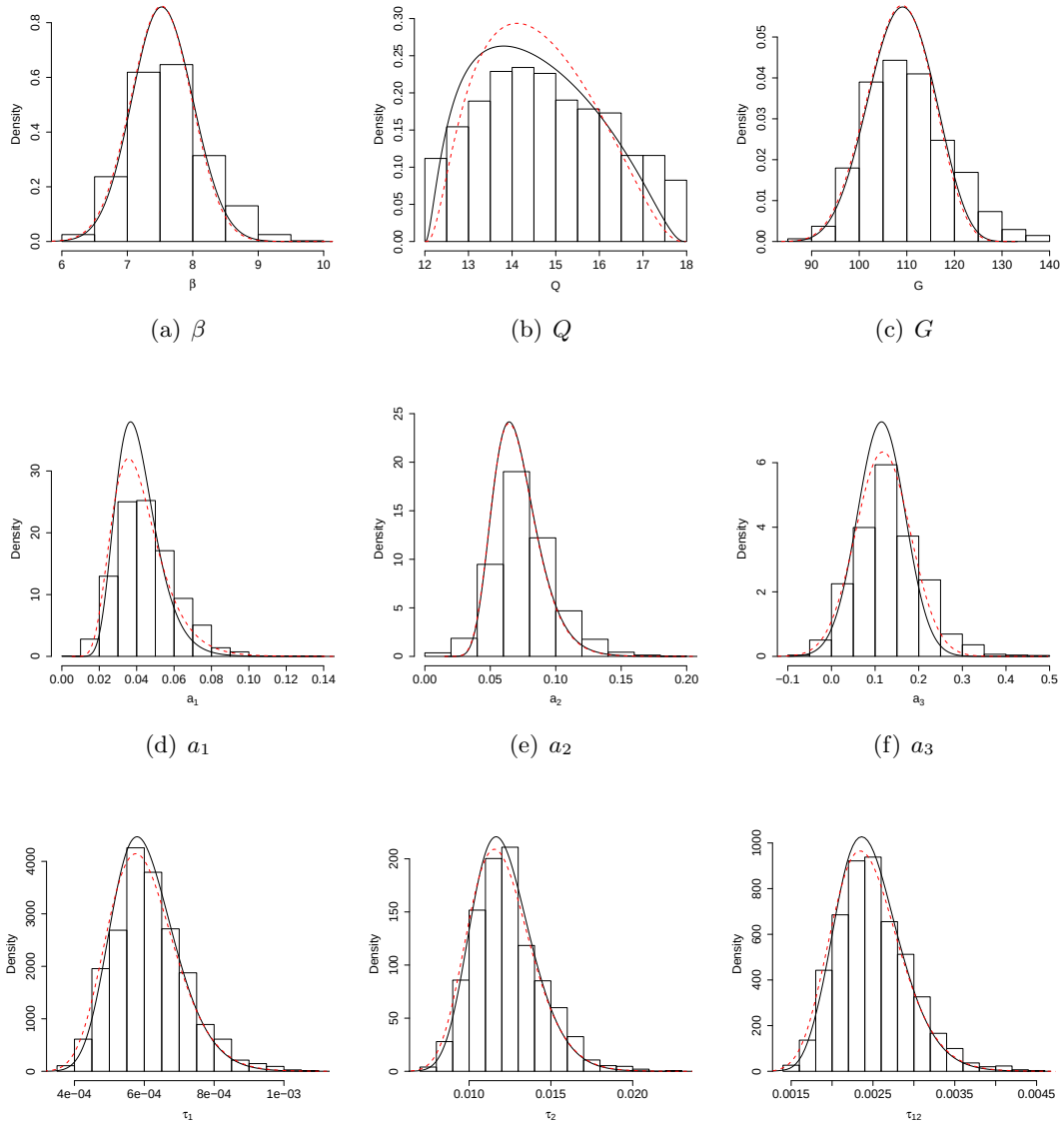
From Table 4.6 we see that the estimates for almost all the PBBM parameters using BCLT and APFF are fairly similar to the ones when using MCMC. Only for the parameters G and a_3, the estimates using BCLT and APFF seem to be slightly different from the ones using MCMC. However, the credible intervals using BCLT and APFF are narrower than the credible intervals when using MCMC. Moreover, it is worth noting that BCLT and APFF perform considerably well in the estimation of the τ_i's. These facts are also reflected in Figure 4.1, which shows the estimated marginal densities for each parameter using MCMC (histogram), APFF (solid line) and BCLT (dashed line).

We now check how INLA performs in terms of computing the DIC and GRS. Table 4.7 shows these goodness-of-fit measures by the computational approach. From

Table 4.7: DIC and GRS - Synthetic Data

	DIC	p_D	D	GRS
MCMC	-757.311	26.710	-784.020	991.726
INLA	-760.965	25.927	-786.892	995.991

Table 4.7 we conclude that INLA performs well in computing DIC and GRS. Moreover, it is intuitive that INLA and MCMC estimate $y_i^{\text{rep}}(t_j)\,|\,\mathbf{y}$'s similarly due to the fact that INLA and MCMC provide similar GRS measures. To confirm our intuition, we



(a) β (b) Q (c) G

(d) a_1 (e) a_2 (f) a_3

(g) τ_1 (h) τ_2 (i) τ_{12}

Figure 4.1: MCMC marginal density estimate (histogram), INLA marginal density estimate: APFF (solid line) and BCLT (dashed line) - Synthetic data.

show in 4.2, the means, 2.5% and 97.5% quantiles for the replicated data computed using MCMC against the same corresponding measures computed using INLA. The solid

line (with slope 1) represents equality between INLA and MCMC in estimating each of these measures. Notice that the dots lie very close to the solid lines, which indicates that

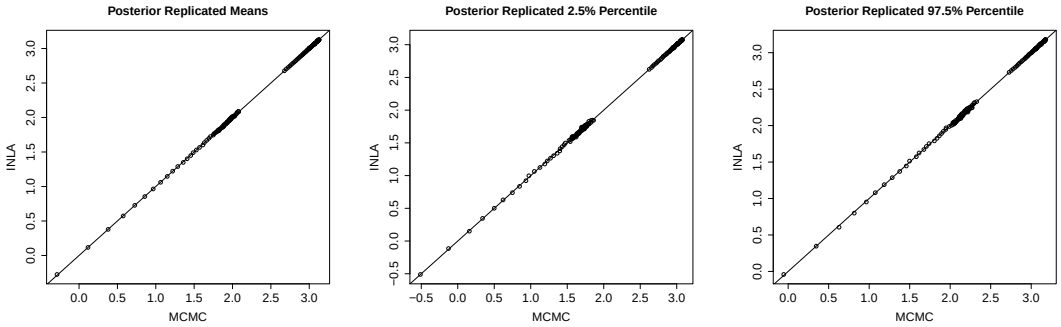


Figure 4.2: MCMC vs INLA in estimating mean, 2.5% and 97.5% quantiles of $y_i^{\text{rep}}(t_j)|\mathbf{y} \ \forall \ i = \{1, 2\}, j = \{1, 2, \ldots, 100\}$ - Synthetic data.

MCMC and INLA provide similar values for the mean and quantiles of the $y_i^{\text{rep}}(t_j) \,|\, \mathbf{y}$'s.

Lastly, we compare INLA and MCMC in terms of computational time. Both methods were coded in R, and shared functions when possible (e.g., solution of the two-zone model, computation of the posterior matrix of $\boldsymbol{\eta}$, evaluation of $p\,(\boldsymbol{\kappa} \,|\, \mathbf{y})$). Table 4.8 shows the computational time to run the PBBM on a Linux server with configuration 2× AMD Opteron 6172 @ 2.10 GHz (24 cores), 16GB of RAM. From this table, we

Table 4.8: Computational Time to Run PBBM using MCMC and INLA in a Linux Server - Synthetic Data

Inference Approach	Marginal Approach	Elapsed Time (s)
MCMC	–	4,122.45
INLA	BCLT	19.693
	APFF	23.752

clearly see that INLA is much faster than MCMC. Regardless of the approach used to estimate the marginal distributions of the parameters in PBBM, INLA is approximately 200 times faster than MCMC.

Now, we compare the PBBM running times considering a Windows machine and not parallelizing the INLA algorithm. In particular, Table 4.9 shows the computational

time to run the PBBM in a PC Intel(R) Core(TM)2 Duo P8600 @ 2.40 GHz and 4GB of RAM. From Table 4.9 we see that even in the worst case scenario INLA is about 30

Table 4.9: Computational Time to Run PBBM using MCMC and INLA in a Windows Machine - Synthetic Data

Inference Approach	Marginal Approach	Elapsed Time (s)
MCMC	–	2,416.55
INLA	BCLT	26.96
	APFF	78.44

times faster than MCMC. Since we showed previously that there is no difference between APFF and BCLT in terms of estimating PBBM parameters, BCLT is the better option.

4.3.2 Experimental Data

In this section we present a similar analysis as before, but using the experimental two-zone data set presented in Section 3.2. Table 4.10 presents the adopted prior distributions for the PBBM. Moreover, we fixed $\phi_1 = \phi_2 = 3.25$ and $\nu_1 = \nu_2 = 2.5$. Table 4.11

Table 4.10: Prior Distributions - Experimental Data

Parameter	Prior Distribution
β	$\mathrm{U}(0, 13)$
Q	$\mathrm{U}(13.7, 13.9)$
G	$\mathrm{U}(351, 352)$
a_1	$\mathrm{LN}(-4.5, 2.1)$
a_2	$\mathrm{LN}(-3.4, 2.1)$
a_3	$\mathrm{N}(0.3, 3)$
$\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}_3)$	$\mathrm{IW}\left(\begin{bmatrix} 0.0226 & 0.0715 \\ 0.0715 & 0.2359 \end{bmatrix}, 5\right)$

shows the potential scale reduction factor for each parameter, computed using 3 parallel MCMC chains with different starting points. Since the 97.5% quantiles are close to 1, we conclude that there is no sign of non-convergence of the MCMC chain.

Table 4.12 presents for each parameter in the PBBM its estimated posterior mean

Table 4.11: Potential scale reduction factors

Parameter	Point est.	97.5% quantile
β	1.00	1.01
Q	1.01	1.02
G	1.00	1.00
a_1	1.00	1.01
a_2	1.01	1.02
a_3	1.00	1.01
τ_1	1.00	1.00
τ_2	1.00	1.00
τ_{12}	1.00	1.00

and the 95% credible interval by the computational approach. The numbers in the square brackets are the corresponding MCSEs. From this Table we conclude that in general the PBBM parameters estimates using BCLT and APFF are reasonable. More specifically, when estimating τ_i's, BCLT and APFF return estimates very close the ones that MCMC provides. However, when estimating, β, a_2 and a_3, BCLT and APFF seem slightly different from MCMC. In addition, the credible intervals using BCLT and APFF are narrower than credible intervals using MCMC. We can also view these facts in the plots in Figure 4.3.

Table 4.12: Posterior summaries for parameters in the PBBM - Real Data

$\boldsymbol{\theta}$	MCMC	APFF	BCLT
β	2.6 (1.9, 3.9) [0.02]	2.5 (2.0, 3.2)	2.5 (1.8, 3.4)
Q	13.8 (13.7, 13.9) [0.003]	13.8 (13.7, 13.9)	13.8 (13.7, 13.9)
G	351.5 (351.0, 352.0) [0.014]	351.5 (351.0, 352.0)	351.5 (351.1, 352.0)
a_1	0.31 (0.25, 0.38) [0.002]	0.30 (0.25, 0.36)	0.30 (0.24, 0.36)
a_2	0.38 (0.25, 0.57) [0.004]	0.36 (0.24, 0.52)	0.35 (0.23, 0.52)
a_3	0.56 (0.32, 0.80) [0.004]	0.59 (0.41, 0.77)	0.59 (0.38, 0.80)
τ_1	0.00013 (0.00011, 0.00015) [2.8e-07]	0.00013 (0.00011, 0.00015)	0.00013 (0.00011, 0.00015)
τ_2	0.00105 (0.00090, 0.00125) [2.2e-06]	0.00105 (0.00090, 0.00123)	0.00105 (0.00089, 0.00123)
τ_{12}	0.00027 (0.00023, 0.00033) [6.6e-07]	0.00027 (0.00022, 0.00032)	0.00027 (0.00022, 0.00032)

Table 4.13 shows the DIC and the GRS computed using INLA and MCMC. As in the previous section, INLA and MCMC return relatively close estimates for DIC and

GRS measures. The figures in 4.4 are the analogues of figures in 4.2 for the real data.

Table 4.13: DIC and GRS - Real Data

	DIC	p_D	$\bar{D}$	GRS
INLA	- 2750.477	79.420	-2829.897	3794.771
MCMC	-2785.589	79.094	-2864.683	3784.725

Finally, we compare MCMC and INLA in terms of running time. Tables 4.14 and 4.15 show the computational times when running the PBBM for the experimental data in the Linux server and Windows machine mentioned in the previous section, respectively.

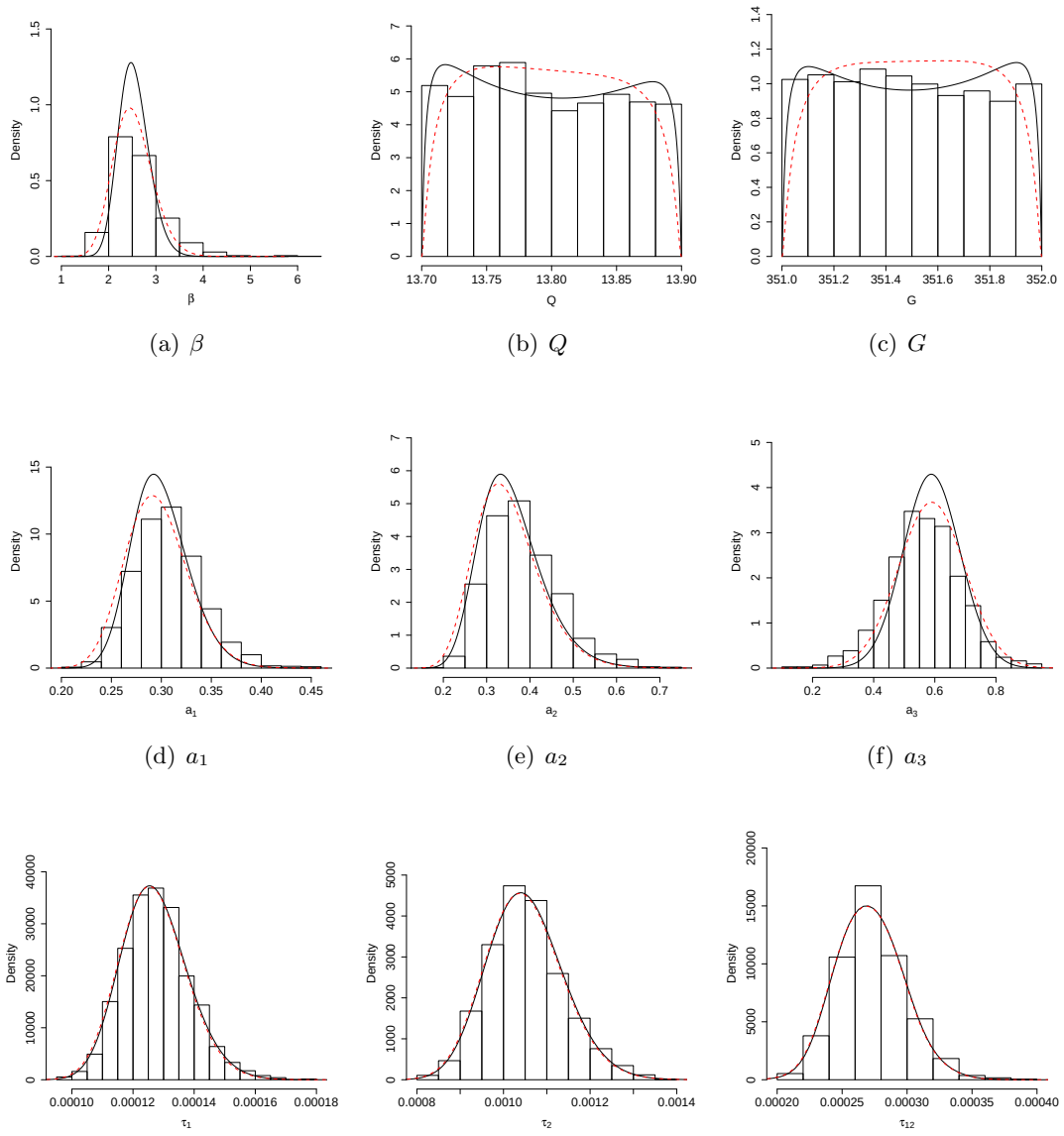
Table 4.14: Computational Time to Run PBBM using MCMC and INLA in a Linux Server - Experimental Data

Inference Approach	Marginal Approach	Elapsed Time (s)
MCMC	–	48,719.2
INLA	BCLT	144.153
	APFF	157.642

Table 4.15: Computational Time to Run PBBM using MCMC and INLA in a Windows Machine - Experimental Data

Inference Approach	Marginal Approach	Elapsed Time (s)
MCMC	–	32,934.5
INLA	BCLT	127.05
	APFF	156.81

From these tables we see that INLA is much faster than MCMC. Moreover, in comparison to the previous section, we conclude that INLA suffers less from the size of the data set than MCMC. In particular, $\mathbf{y}$ had length 200 in the previous section and took 23.752 and 4,122.45 seconds to run the PBBM to obtain complete inferential output using INLA and MCMC, respectively. In this section $\mathbf{y}$ has length 511, and took



(a) β (b) Q (c) G

(d) a_1 (e) a_2 (f) a_3

(g) τ_1 (h) τ_2 (i) τ_{12}

Figure 4.3: MCMC marginal density estimate (histogram), INLA marginal density estimate (solid line) and normal approximation (dashed line) - Experimental data.

157.642 and 48,719.2 seconds to run the PBBM using INLA and MCMC, respectively. Therefore, while the time for MCMC increased approximately 12 times, the time for

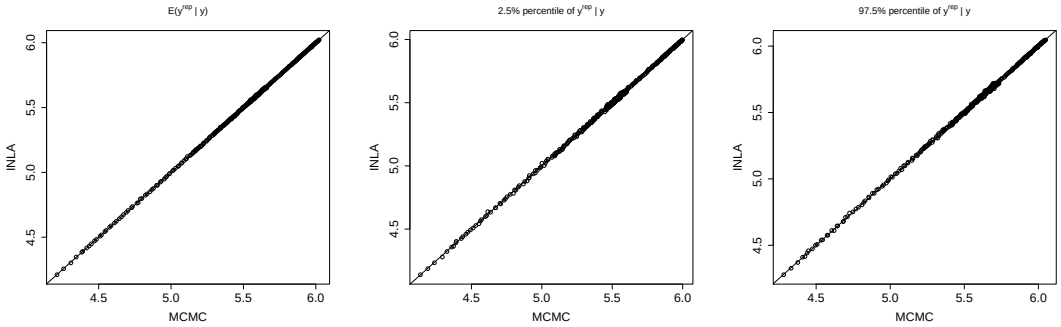


Figure 4.4: MCMC vs INLA according to the mean, 2.5% and 95% of $y_i^{\text{rep}}|\mathbf{y} \quad \forall \quad i = 1, 2, \ldots, 511$ - Experimental data.

INLA increased only approximately 6.6 times.

4.4 Discussion

In this chapter we compared INLA to the conventional MCMC sampling method in the context of the process-based Bayesian Melding framework applied to two-zone models. In our comparison studies, we found that INLA can provide, in general, reasonable estimates for PBBM parameters, good approximations for the predictive distribution and goodness-of-fit measures such as Deviance Information Criterion (DIC) and Gneiting-Raftery Scoring Rule (GRS). Moreover, INLA is dramatically faster than MCMC, in some cases by an order of 200 times.

One limitation of applying INLA in to the PBBM is that we had to fix the Matérn parameters ϕ_i's and ν_i's. This makes the PBBM less flexible, and consequently favoring loss in the quality of the fit. One strategy would be fit the PBBM using INLA and plot the means for the replicated data against the observed log-exposure concentrations, with a solid line (with slope 1) that represents equality between the model replicated means and the observations. If this plot shows dots close to the solid line, the fit using INLA is sufficient. If not, fit the PBBM using MCMC considering ϕ_i's and ν_i's not fixed.

Lastly, MCMC has wider applicability and is, in general, easier to design and implement. In addition, it will be superior to INLA when the number of hyperparameters is large – Bayesian inference in the sciences are no longer restricted to models with a handful of hyperparameters. Finally, unlike MCMC, INLA is not sampling-based in a natural way. The ability to sample from the posterior endows the statistician with a diverse set of tools for model assessment and comparison. The books by [18] and [49] outline numerous sampling-based techniques for conducting posterior predictive checks and assessment and comparison of competing models. On the other hand, computing DIC and posterior predictive distributions is clumsier in INLA because they involve high-dimensional integrals. Finally, while MCMC and INLA will both require tuning specific to the data at hand, the design of software implementing these methods will be quite different. While MCMC software will build upon graphical models, as is done in BUGS and JAGS, INLA is perhaps best offered to the end-user as suite of wrapper functions in statistical languages such as R or MATLAB.

Chapter 5

Conclusion

In this thesis we presented and discussed a process-based Bayesian melding (PBBM) of occupational exposure models and field data. Based upon our current findings, we advocate estimating inputs to the physical model whenever possible. We recognize that full inference will require solving the physical model, which may be infeasible in certain settings. However, a very large number of physical processes can be formulated as a general systems of linear ODE's, whose solutions closely depend upon the eigenvalues of the coefficient matrix (see Appendix A). Assigning reasonable priors to the eigenvalues will yield tractable solutions to such systems making the PBBM framework widely applicable.

Our results also show that approximate inference for the PBBM using INLA provides useful results and is much faster than the conventional MCMC sampling method. However, it is important to mention that this approximation is based on the assumption that the Matérn parameters are fixed. Our next step is to provide a graphical user interface (GUI) software implementing the PBBM, where the user will have the option to choose fitting the lighter version of the PBBM using INLA, or fitting the full version of the PBBM using MCMC.

We present in the following sections a few extensions of the PPBM that are under exploration.

5.1 Space-time measurements

The turbulent eddy-diffusion model is a physical model that accounts for a continuous concentration gradient outward from the source, unlike two-zone models. Turbulent diffusion refers to the idea of an eddy, or swirl, of room air grabbing a parcel of contaminant at one location and carrying it to another location. This macro-scale random motion of air parcels leads to the dispersion of contaminant away from the source. A key parameter is the turbulent eddy diffusion coefficient D_T (m^2/min). Using this model requires specifying the worker's location relative to the source. The concentration as a function of location $\mathbf{s}$ (in 3-dimensional Euclidean coordinates) and time t is given by

$$c(\mathbf{s},t) = \frac{G}{2\pi D_T \|\mathbf{s}\|}\left\{1 - \mathrm{erf}\left(\frac{\|\mathbf{s}\|}{\sqrt{4D_T t}}\right)\right\}, \tag{5.1}$$

where $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z \exp\left(-u^2\right) du$, G is the contaminant generation rate, $\mathbf{s}$ represents the 3-dimensional coordinate location of the worker with the source assumed to be located at the origin, $\|\mathbf{s}\| > 0$ and t represents time.

Therefore, having measured exposure concentrations in different spatial locations at each time point, we could extend the PBBM by combining the turbulent eddy-diffusion model with spatio-temporal stochastic processes to create highly flexible melding frameworks, i.e.,

$$y(\mathbf{s},t) = f(\boldsymbol{\theta}_1; \mathbf{x}, \mathbf{s}, t) + \eta(\mathbf{s},t) + \epsilon(\mathbf{s},t)$$

where $f(\boldsymbol{\theta}_1; \mathbf{x}, \mathbf{s}, t) = \log c(\mathbf{s},t)$, $\boldsymbol{\theta}_1 = \{D_T, G\}$.

5.2 Controlled inputs

Consider the following situation: a hygienist conducts multiple independent "runs" of the two-zone model corresponding to a set of inputs. Here, the input parameter $\boldsymbol{\theta}_1$ is considered controlled, i.e, known with precision. The objective, therefore, is no longer to estimate $\boldsymbol{\theta}_1$ but to synthesize, or meld, the two-zone model outputs from these controlled inputs with the observed data to construct an efficient predictive framework.

Assume the hygienist chooses r sets of controlled inputs, say $\{\boldsymbol{\theta}_{1j}\}$, $j = 1, \ldots, r$. The number of sets, r, can be as small as one or as large as to reasonably cover the

plausible range of inputs. For each combination of time points and inputs, i.e. t_i and $\boldsymbol{\theta}_{1j}$, the hygienist runs the two-zone model to produce an ensemble of outputs, $\mathbf{z}^j(t_i) = \mathbf{f}(\boldsymbol{\theta}_{1j}; \mathbf{x}, t_i)$, $i = 1, \ldots, n$. Our "data" then comprises the observed $\mathbf{y}(t_i)$'s and the output ensemble, $\mathbf{z}^j(t_i)$'s, which we denote by $\mathbf{y}$ and $\mathbf{z}$ respectively.

As in Section 3.3, we propose process-based models, but now one for each observed outcome and each run. To be precise,

$$\mathbf{y}(t) = \boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\alpha}_{\mathbf{y}}; t) + \boldsymbol{\eta}_{\mathbf{y}}(t) + \boldsymbol{\epsilon}_{\mathbf{y}}(t); \tag{5.2}$$

$$\mathbf{z}^j(t) = \boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\alpha}_{\mathbf{y}}; t) + \boldsymbol{\eta}_{\mathbf{z}^j}(t) + \boldsymbol{\epsilon}_{\mathbf{z}^j}(t), \quad j = 1, \cdots, r \tag{5.3}$$

where $\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\alpha}_{\mathbf{y}}; t)$ is a parametric function of t, such as a regression with slopes $\boldsymbol{\alpha}_{\mathbf{y}}$; $\boldsymbol{\eta}_{\mathbf{y}}(t)$ is a stochastic process complementing $\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\alpha}_{\mathbf{y}}; t)$ by capturing small-scale variation due to unaccounted associations and complexities, and $\boldsymbol{\epsilon}_{\mathbf{y}}(t)$ is a random white-noise process that captures variation due to measurement error. The structure for each $\mathbf{z}^j(t)$ is similar with notations akin to those for the $\mathbf{y}(t)$.

Clearly, we need to connect (5.2) and (5.3), since we seek to borrow strength from the $\mathbf{z}^j(t)$'s for predicting $\mathbf{y}(t)$. Our approach is to suppose that the processes $\boldsymbol{\eta}_{\mathbf{y}}(t)$ and $\boldsymbol{\eta}_{\mathbf{z}^j}(t)$ have a "true" common underlying process, say $\mathbf{w}(t)$. This can be achieved using linear transformations $\boldsymbol{\eta}_{\mathbf{y}}(t) = \mathbf{L}_{\mathbf{y}}\mathbf{w}(t)$ and $\boldsymbol{\eta}_{\mathbf{z}^j}(t) = \mathbf{L}_{\mathbf{z}^j}\mathbf{w}(t)$, where $\mathbf{L}_{\mathbf{y}}$ and $\mathbf{L}_{\mathbf{z}^j}$ are 2×2 lower triangular matrices (with unknown entries) related to the dispersion structures posited for $\boldsymbol{\eta}_{\mathbf{y}}(t)$ and $\boldsymbol{\eta}_{\mathbf{z}^j}(t)$.

Before introducing the stochastic assumptions, we still have to define the form of $\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\alpha}_{\mathbf{y}}; t)$ and $\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\alpha}_{\mathbf{z}}; t)$. Customarily the mean levels are polynomial functions of some regressors (in our case t) with coefficients $\boldsymbol{\alpha}_{\mathbf{y}}$ and $\boldsymbol{\alpha}_{\mathbf{z}}$. For illustration purposes, here we only consider mean levels that are intercept coefficients. Specifically, we utilize $\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\alpha}_{\mathbf{y}}; t) = \boldsymbol{\alpha}_{\mathbf{y}} = [\alpha_{yn} \;\; \alpha_{yf}]^{\mathrm{T}}$ and $\boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\alpha}_{\mathbf{z}}; t) = \boldsymbol{\alpha}_{\mathbf{z}} = [\alpha_{zn} \;\; \alpha_{zf}]^{\mathrm{T}}$. Therefore, equations (5.2) and (5.3) become

$$\begin{aligned} \mathbf{y}(t) &= \boldsymbol{\alpha}_{\mathbf{y}} + \mathbf{L}_{\mathbf{y}}\mathbf{w}(t) + \boldsymbol{\epsilon}_{\mathbf{y}}(t); \\ \mathbf{z}^j(t) &= \boldsymbol{\alpha}_{\mathbf{z}} + \mathbf{L}_{\mathbf{z}^j}\mathbf{w}(t) + \boldsymbol{\epsilon}_{\mathbf{z}^j}(t), \quad j = 1, \cdots, r. \end{aligned}$$

The probability distribution for the components in $\mathbf{y}(t)$ and $\mathbf{z}^j(t)$'s, as well as the

notations for the parameters, are analogous to those for the PBBM (Section 3.3)

$$\begin{aligned}
\mathbf{w}(t) &\sim \mathrm{GP}_2\left(\mathbf{0}_2, \mathbf{C}_{\mathbf{w}}(\boldsymbol{\varphi}; \cdot, \cdot)\right); \\
\boldsymbol{\epsilon}_{\mathbf{y}}(t_i) \,|\, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_y}(\boldsymbol{\theta}_{3y}) &\overset{\text{iid}}{\sim} \mathrm{N}_2\left(\mathbf{0}_2, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_y}(\boldsymbol{\theta}_{3y})\right),\ i = 1, 2, \ldots, n; \\
\boldsymbol{\epsilon}_{\mathbf{z}_j}(t_i) \,|\, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_z}(\boldsymbol{\theta}_{3z}) &\overset{\text{iid}}{\sim} \mathrm{N}_2\left(\mathbf{0}_2, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_z}(\boldsymbol{\theta}_{3z})\right), i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, r\ ,
\end{aligned}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_y}(\boldsymbol{\theta}_{3y})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_z}(\boldsymbol{\theta}_{3z})$ are 2×2 covariance matrices that are functions of unknown parameters $\boldsymbol{\theta}_{3y}$ and $\boldsymbol{\theta}_{3z}$ respectively.

The cross-covariance $\mathbf{C}_{\mathbf{w}}(\boldsymbol{\varphi}; \cdot, \cdot)$ is computed exactly as in Section 3.3, and therefore $\boldsymbol{\varphi} = \{\phi_1, \phi_2, \nu_1, \nu_2\}$. However, for this model, $\boldsymbol{\theta}_2$ does not contain $\boldsymbol{\varphi}$, i.e, $\boldsymbol{\theta}_2 = \{\mathbf{L}_{\mathbf{y}}, \mathbf{L}_{\mathbf{z}^1}, \ldots, \mathbf{L}_{\mathbf{z}^r}\}$ in order to write the posterior distribution more parsimoniously. The measurement errors at the near and far fields can be also considered correlated or uncorrelated. For the former,

$$\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_y}(\boldsymbol{\theta}_{3y}) = \begin{bmatrix} \tau_1 & \tau_{12} \\ \tau_{12} & \tau_2 \end{bmatrix} \text{ and } \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_z}(\boldsymbol{\theta}_{3z}) = \begin{bmatrix} \delta_1 & \delta_{12} \\ \delta_{12} & \delta_2 \end{bmatrix},$$

thus $\boldsymbol{\theta}_3 = \{\boldsymbol{\theta}_{3y}, \boldsymbol{\theta}_{3z}\} = \{\tau_1, \tau_{12}, \tau_2, \delta_1, \delta_{12}, \delta_2\}$. For the latter, $\tau_{12} = \delta_{12} = 0$, which makes $\boldsymbol{\theta}_3 = \{\tau_1, \tau_2, \delta_1, \delta_2\}$. Assuming that $\boldsymbol{\epsilon}_{\mathbf{y}}(t_i)$'s and $\boldsymbol{\epsilon}_{\mathbf{z}_j}(t_i)$'s are independent, the joint posterior distribution of $\boldsymbol{\theta} = \{\boldsymbol{\alpha}_{\mathbf{y}}, \boldsymbol{\alpha}_{\mathbf{z}}, \boldsymbol{\varphi}, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ and $\mathbf{w} = \left[\mathbf{w}^{\mathrm{T}}(t_1) \cdots \mathbf{w}^{\mathrm{T}}(t_n)\right]^{\mathrm{T}}$ is given by

$$\begin{aligned}
p\left(\boldsymbol{\theta}, \mathbf{w} \,|\, \mathbf{y}, \mathbf{z}, \mathbf{t}, \boldsymbol{\gamma}\right) \propto \prod_{i=1}^{n} \Big\{ & \mathrm{N}_2\left(\mathbf{y}(t_i) \,|\, \boldsymbol{\alpha}_{\mathbf{y}} + \mathbf{L}_{\mathbf{y}}\mathbf{w}(t_i), \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_y}(\boldsymbol{\theta}_{3y})\right) \\
& \times \prod_{j=1}^{r} \mathrm{N}_2\left(\mathbf{z}_j(t_i) \,|\, \boldsymbol{\alpha}_{\mathbf{z}} + \mathbf{L}_{\mathbf{z}^j}\mathbf{w}(t_i), \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_z}(\boldsymbol{\theta}_{3z})\right) \Big\} \\
& \times \mathrm{N}_{2n}\left(\mathbf{w} \,|\, \mathbf{0}_{2n}, \boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\varphi}; \mathbf{t})\right) \times p(\boldsymbol{\theta} \,|\, \boldsymbol{\gamma})\ , \qquad (5.4)
\end{aligned}$$

where $\boldsymbol{\gamma}$ is the collection of all hyperparameters.

To complete the Bayesian hierarchical model we assign the following independent

prior distributions:

$$\boldsymbol{\alpha}_{\mathbf{y}} \sim \mathrm{N}_2(\mathbf{m}_{\boldsymbol{\alpha}_y}, \mathbf{S}_{\boldsymbol{\alpha}_y}) \text{ and } \boldsymbol{\alpha}_{\mathbf{z}} \sim \mathrm{N}_2(\mathbf{m}_{\boldsymbol{\alpha}_z}, \mathbf{S}_{\boldsymbol{\alpha}_z});$$

$$\tau_l \sim \mathrm{IG}(a_{\tau_l}, b_{\tau_l}) \text{ and } \delta_l \sim \mathrm{IG}(a_{\delta_l}, b_{\delta_l}), \quad l = \{N, F\}; \tag{5.5}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_y}(\boldsymbol{\theta}_{3y}) \sim \mathrm{IW}(\mathbf{m}_{\boldsymbol{\epsilon}_y}, \Psi_{\boldsymbol{\epsilon}_y}) \text{ and } \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_z}(\boldsymbol{\theta}_{3z}) \sim \mathrm{IW}(\mathbf{m}_{\boldsymbol{\epsilon}_z}, \Psi_{\boldsymbol{\epsilon}_z}), \tag{5.6}$$

where the prior distributions in (5.5) and (5.6) are for a model with uncorrelated and correlated error measurements, respectively.

The prior distributions for the entries of $\mathbf{L}_{\mathbf{y}}$ and $\mathbf{L}_{\mathbf{z}^j}$'s are defined just like for $\mathbf{A}$ in Section 3.3, i.e., log-normal distributions for each entry in the diagonal and normal distributions for the off-diagonal elements. Likewise, the remaining parameters in $\boldsymbol{\varphi}$ are assigned the prior distributions: $\phi_i \sim \mathrm{U}(a_{\phi_i}, b_{\phi_i})$, and $\nu_i \sim \mathrm{U}(a_{\nu_i}, b_{\nu_i})$, for $i = 1, 2$.

We did not find this model had an advantage over a simple Bayesian model such as $\mathbf{y}(t) = \boldsymbol{\mu}(t) + \boldsymbol{\eta}(t) + \boldsymbol{\epsilon}(t)$, but we are still exploring scenarios which might benefit from this model.

5.3 Dynamic physical inputs

Lastly, another extension would allow the physical inputs vary over time. For example, in the two-zone setting, we would decompose the observed log-exposure at time t as $\mathbf{y}(t) = \mathbf{f}(\boldsymbol{\theta}(t); \mathbf{x}, t) + \boldsymbol{\eta}(t) + \boldsymbol{\epsilon}(t)$, where $\boldsymbol{\theta}(t) = \{\beta(t), Q(t), G(t)\}$. That would improve the physical model's performance and return information of how the physical system behaves during a period of time. The challenge in this model is to assume a correlation structure of $\boldsymbol{\theta}(t)$ such that the data is strong enough to successfully estimate it.

Chapter 6

References

[1] Mark Nicas. Estimating exposure intensity in an imperfectly mixed room. *American Industrial Hygiene Association Journal*, 57:542–550, Jun 1996.

[2] Gurumurthy Ramachandran. *Occupational Exposure Assessment for Air Contaminants*. CRC Press, Taylor & Francis Group, 2005.

[3] Charles B. Keil, Wil F. ten Berge, M. Cathy Fehenbacher, Michael A. Jayjock, Mark Nicas, and Patricia Reinke. *Mathematical Models for Estimating Occupational Exposure to Chemicals*. American Industrial Hygiene Association, second edition, 2009.

[4] Gurumurthy Ramachandran. Retrospective exposure assessment using Bayesian methods. *Annals of Occupational Hygiene*, 45(8):651–667, 2001.

[5] Paul Hewett, Perry Logan, John Mulhausen, Gurumurthy Ramachandran, and Sudipto Banerjee. Rating exposure control using Bayesian decision analysis. *Journal of Occupational and Environmental Hygiene*, 3, 2006.

[6] Pierre-Edouard Sottas, Jérome Lavoué, Raffaella Bruzzi, David Vernez, Nicole Charrière, and Pierre-Olivier Droz. An empirical hierarchical Bayesian unification of occupational exposure assessment methods. *Statistics in Medicine*, 28:75–93, 2009.

[7] Yufen Zhang, Sudipto Banerjee, Rui Yang, Claudiu Lungu, and Gurumurthy Ramachandran. Bayesian modeling of exposure and air flow using two-zone models. *The Annals of Occupational Hygiene*, 53(4):409–424, 2009.

[8] Adrian E. Raftery, Geof H. Givens, and Judith E. Zeh. Inference from a deterministic population dynamics model for bowhead whales. *Journal of the American Statistical Association*, 90:402–416, 1995.

[9] David Poole and Adrian E. Raftery. Inference for deterministic simulation models: The Bayesian melding approach. *Journal of the American Statistical Association*, 95:1244–1255, Dec 2000.

[10] Montserrat Fuentes and Adrian E. Raftery. Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. *Biometrics*, 61:36–45, 2005.

[11] Hana Ševčíková, Adrian E. Raftery, and Paul A. Waddell. Assessing uncertainty in urban simulations using Bayesian melding. *Transportation Research Part B*, 41:652–669, 2007.

[12] Hana Ševčíková, Adrian E. Raftery, and Paul A. Waddell. Uncertain benefits: Application of Bayesian melding to the Alaskan way viaduct in Seattle. *Transportation Research Part A*, 45:540–553, 2011.

[13] Adrian E. Raftery and Le Bao. Estimating and projecting trends in HIV/AIDS generalized epidemics using incremental mixture importance sampling. *Biometrics*, 66:1162–1173, Dec 2010.

[14] Veronica J. Berrocal, Alan E. Gelfand, and David M. Holland. Space-time data fusion under error in computer model output: An application to modeling air quality. *Biometrics*, 2011.

[15] Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.

[16] María J. Bayarri, James O. Berger, Rui Paulo, Jerry Sacks, John A. Cafeo,

J. Cavendish, C. H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49:138–154, 2007.

[17] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[18] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition, 2003.

[19] Herman K. Van Dijk, J. Peter Hop, and Adri S. Louter. An algorithm for the computation of posterior moments and densities using simple importance sampling. *The Statistician*, 36:83–90, 1987.

[20] Donald B. Rubin. The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398):543–546, Jun 1987.

[21] Donald B. Rubin. Using the SIR algorithm to simulate posterior distributions. In J. M. Bernardo, M. H. Degroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics 3*. Oxford University Press, 1988.

[22] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 2004.

[23] Russell J. Steele, Adrian E. Raftery, and Mary J. Emond. Computing normalizing constants for finite mixture models via incremental mixture importance sampling (IMIS). *Journal of Computational and Graphical Statistics*, 15(3):712–734, 2006.

[24] Adrian E. Raftery and Le Bao. Estimating and projecting trends in HIV/AIDS generalized epidemics using incremental mixture importance sampling. *Biometrics*, 66:1162–1173, Dec 2010.

[25] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and

the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[26] Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, Jun 1990.

[27] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal Chemical Physics*, 21(6):1087–1092, Jun 1953.

[28] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, Apr 1970.

[29] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.

[30] David M. Gay. Usage summary for selected optimization routines. Technical Report 153, AT & T Bell Laboratories, Department of Computing Science, 1990.

[31] Paul Gilbert. **numDeriv***: Accurate Numerical Derivatives*, 2011. R package version 2010.11-1.

[32] Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

[33] Stephen P. Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, Dec 1998.

[34] Sudipto Banerjee, Bradley P. Carlin, and Alan E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall/CRC, 2004.

[35] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, ninth edition, 1965.

[36] Michael L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, New York, 1999.

[37] David J. Spiegelhalter, Nicola G. Best, Bradley P. Carlin, and Angelika van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society B*, 64:583–639, 2002.

[38] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[39] James Marshall Flegal. *Monte Carlo Standard Errors for Markov Chain Monte Carlo*. PhD thesis, University of Minnesota, Minneapolis MN, 2008.

[40] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

[41] Haario Heikki, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[42] Gareth O. Roberts and Jeffrey S. Rosenthal. Example of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

[43] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, Cambridge MA, 2001.

[44] Sara Martino. *Approximate Bayesian Inference for Latent Gaussian Models*. PhD thesis, Norwegian University of Science and Technology, Trondheim, 2007.

[45] Havard Rue and Sara Martino. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society B*, 71:319–392, 2009.

[46] Jarno Vanhatalo, Ville Pietiläinen, and Aki Vehtari. Approximate inference for disease mapping with sparse Gaussian processes. *Statistics in Medicine*, 29(15):1580–1607, 2010.

[47] G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society B*, 13:1–45, 1951.

[48] S. M. Sanchez and P. J. Sanchez. Very large fractional factorials and central composite designs. *ACM Transactions on Modeling and Computer Simulation*, 15:362–377, 2005.

[49] Bradley P. Carlin and Thomas A. Louis. *Bayesian Methods for Data Analysis*. Chapman & Hall/CRC, New York, third edition, 2008.

Appendix A

Although the current application considered a two-dimensional physical model, our proposed framework applies to m-dimensional linear systems of ODE's. Consider the general linear system

$$\frac{d}{dt}\mathbf{c}(t) = \mathbf{W}\mathbf{c}(t) + \mathbf{g}\ , \tag{A.1}$$

where $\mathbf{c}(t)$ is an $m \times 1$ vector function of t, $\mathbf{W}$ is an $m \times m$ matrix (which may depend upon known and unknown inputs but we suppress them in the notation here) that has m real and distinct eigenvalues and $\mathbf{g}$ is a vector of length m.

When $\mathbf{W}$ has m distinct eigenvalues, we can find a non-singular matrix $\mathbf{P}$ such that $\mathbf{W} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$, where the columns of $\mathbf{P}$ are linearly independent eigenvectors. Therefore, $e^{\mathbf{W}} = \mathbf{P}e^{\mathbf{\Lambda}}\mathbf{P}^{-1}$, where $\mathbf{\Lambda}$ is a diagonal matrix with λ_i as the i-th diagonal element, $i = \{1, 2, \cdots, m\}$. Now, let $\mathbf{G}_i = \mathbf{u}_i\mathbf{v}_i^{\mathrm{T}}$, where $\mathbf{u}_i$ is the i-th column of $\mathbf{P}$ and $\mathbf{v}_i^{\mathrm{T}}$ is the i-th row of $\mathbf{P}^{-1}$. (These are often referred to as the *right* and *left* eigenvectors respectively.) It is straightforward to see that (i) $\mathbf{G}_i^2 = \mathbf{G}_i$, (ii) $\mathbf{G}_i\mathbf{G}_j = 0\ \ \forall\ \ i \neq j$ and (iii) $\sum_{i=1}^{m} \mathbf{G}_i = \mathbf{I}_m$. Each $\mathbf{G}_i$ is idempotent and is, in fact, the oblique projector onto the null space of $\mathbf{W} - \lambda_i\mathbf{I}_m$ along the column space of $\mathbf{W} - \lambda_i\mathbf{I}_m$. It is also easily verified that $e^{t\mathbf{W}}e^{-t\mathbf{W}} = \mathbf{I}_m$ and $e^{t\mathbf{W}}\mathbf{W}^{-1} = \mathbf{W}^{-1}e^{t\mathbf{W}}$.

From the above properties of the $\mathbf{G}_i$ matrix, it easily follows that $e^{\mathbf{W}} = \sum_{i=1}^{m} e^{\lambda_i}\mathbf{G}_i$. Consequently,

$$\frac{d}{dt}e^{t\mathbf{W}} = \sum_{i=1}^{m} \lambda_i e^{\lambda_i t}\mathbf{G}_i = \sum_{i=1}^{m} \lambda_i e^{\lambda_i t}\mathbf{u}\mathbf{v}_i^{\mathrm{T}} = \mathbf{P}\mathbf{\Lambda}e^{t\mathbf{\Lambda}}\mathbf{P}^{-1} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}\mathbf{P}e^{t\mathbf{\Lambda}}\mathbf{P}^{-1} = \mathbf{W}e^{t\mathbf{W}}$$

$$\text{and} \quad \int e^{t\mathbf{W}}dt = \sum_{i=1}^{m} \frac{1}{\lambda_i} e^{\lambda_i t}\mathbf{G}_i = \mathbf{P}\mathbf{\Lambda}^{-1}e^{t\mathbf{\Lambda}}\mathbf{P}^{-1} = \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^{-1}\mathbf{P}e^{t\mathbf{\Lambda}}\mathbf{P}^{-1} = \mathbf{W}^{-1}e^{t\mathbf{W}}\ .$$

Multiplying both sides of (A.1) by $e^{-t\mathbf{W}}$ from the left yields:

$$e^{-t\mathbf{W}}\left[\frac{d}{dt}\mathbf{c}(t)-\mathbf{W}\mathbf{c}(t)\right]=e^{-t\mathbf{W}}\mathbf{g}\Longrightarrow\frac{d}{dt}\left[e^{-t\mathbf{W}}\mathbf{c}(t)\right]=e^{-t\mathbf{W}}\mathbf{g}\,. \tag{A.2}$$

Integrating out both sides of (A.2), we obtain $e^{-t\mathbf{W}}\mathbf{c}(t)=-\mathbf{W}^{-1}e^{-t\mathbf{W}}\mathbf{g}+\mathbf{k}$, where $\mathbf{k}$ is a constant vector. The initial condition at $t=0$ yields $\mathbf{c}(0)=-\mathbf{W}^{-1}\mathbf{g}+\mathbf{k}$, so $\mathbf{k}=\mathbf{c}(0)+\mathbf{W}^{-1}\mathbf{g}$. Consequently, $\mathbf{c}(t)=e^{t\mathbf{W}}\mathbf{c}(0)+\mathbf{W}^{-1}\left[e^{t\mathbf{W}}-\mathbf{I}_m\right]\mathbf{g}$ is the solution to (A.1).

The two-zone model (Section 2.2) fits into the above framework with $m=2$. Therefore, to use the result just derived, we have to guarantee that $\mathbf{W}$ has 2 distinct eigenvalues. The eigenvalues of $\mathbf{W}$ determine the numerical stability and the physical interpretability of the two-zone model. The two eigenvalues of $\mathbf{W}$ are the roots of the characteristic polynomial $\lambda^2+\left(\frac{\beta}{V_N}+\frac{\beta+Q}{V_F}\right)\lambda+\frac{\beta Q}{V_NV_F}=0$. Note that $\det(\mathbf{W})=\dfrac{\beta Q}{V_NV_F}$, which means that $\mathbf{W}$ is nonsingular as long as β and Q are not zero. The eigenvalues are available in closed form as

$$\begin{aligned}\lambda_1&=\tfrac{1}{2}\left[-\left(\tfrac{\beta V_F+(\beta+Q)V_N}{V_NV_F}\right)+\sqrt{\left(\tfrac{\beta V_F+(\beta+Q)V_N}{V_NV_F}\right)^2-4\left(\tfrac{\beta Q}{V_NV_F}\right)}\right],\\ \lambda_2&=\tfrac{1}{2}\left[-\left(\tfrac{\beta V_F+(\beta+Q)V_N}{V_NV_F}\right)-\sqrt{\left(\tfrac{\beta V_F+(\beta+Q)V_N}{V_NV_F}\right)^2-4\left(\tfrac{\beta Q}{V_NV_F}\right)}\right].\end{aligned} \tag{A.3}$$

Furthermore, since β, Q, V_F and V_N are all strictly positive, we find

$$\begin{aligned}\left(\frac{\beta V_F+(\beta+Q)V_N}{V_NV_F}\right)^2-4\left(\frac{\beta Q}{V_NV_F}\right)&=\frac{\beta^2V_F^2+2\beta V_F(\beta+Q)V_N+(\beta+Q)^2V_N^2}{(V_NV_F)^2}-4\frac{\beta Q}{V_NV_F}\\ &=\frac{(\beta V_F-QV_N)^2+2\beta^2V_FV_N+(\beta^2+2\beta Q)V_N^2}{(V_NV_F)^2}>0\,.\end{aligned}$$

This implies that the eigenvalues in (A.3) are real and distinct.

Appendix B

Since the marginal distribution of $\mathbf{y}$ is a multivariate normal, we only need check the identifiability of the parameters in the first two moments, i.e., the mean vector and the covariance matrix of $\mathbf{y}$. In particular, we consider the parameters in $\boldsymbol{\Sigma}_{\mathbf{y}}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$. The less obvious situation arises when $\mathbf{A}$ has structure "LT". Then,

$$\mathsf{VAR}\left\{\mathbf{y}\left(t_i\right)\right\} = \begin{bmatrix} a_1^2 & a_1 a_3 \\ a_1 a_3 & a_2^2 + a_3^2 \end{bmatrix} + \begin{bmatrix} \tau_1 & \tau_{12} \\ \tau_{12} & \tau_2 \end{bmatrix}$$

$$\mathsf{COV}\left\{\mathbf{y}\left(t_i\right), \mathbf{y}\left(t_j\right)\right\} = \begin{bmatrix} a_1^2 \rho_1(\varphi_1; t_i, t_j) & a_1 a_3 \rho_1(\varphi_1; t_i, t_j) \\ a_1 a_3 \rho_1(\varphi_1; t_i, t_j) & a_2^2 \rho_2(\varphi_2; t_i, t_j) + a_3^2 \rho_1(\varphi_1; t_i, t_j) \end{bmatrix}$$

where $i, j \in \{1, \cdots, n\}$ and $i \neq j$. Switching a_1^2 and τ_1 will still render the same $\mathsf{VAR}\left\{\mathbf{y}\left(t_i\right)\right\}$ but it will affect $\mathsf{COV}\left\{\mathbf{y}\left(t_i\right), \mathbf{y}\left(t_j\right)\right\}$, which ensures identifiability. The consequence is similar when we switch $a_1 a_3$ and τ_{12}, or $a_2^2 + a_3^2$ and τ_2. Therefore, we can say that the a's and τ's are identifiable. Likewise, if we switch a_k^2 and $\rho_k(\varphi_k; t_i, t_j)$ in $\mathsf{COV}\left\{\mathbf{y}\left(t_i\right), \mathbf{y}\left(t_j\right)\right\}$, it impacts $\mathsf{VAR}\left\{\mathbf{y}\left(t_i\right)\right\}$, where $k = \{1, 2\}$. Therefore, the a's and ρ's are also identifiable.

We point out the need to impose a restriction on the domain of a_2. In fact, $a_2 = x$ and $a_2 = -x$, where $x \neq 0$, returns the same $\mathsf{VAR}\left\{\mathbf{y}\left(t_i\right)\right\}$ and $\mathsf{COV}\left\{\mathbf{y}\left(t_i\right), \mathbf{y}\left(t_j\right)\right\}$. Therefore, to ensure identifiability of this parameter, we restrict it to be greater than 0. Note that when $\mathbf{A}$ has the "D" structure, we also restrict a_1 to be greater than 0. Identifiability conditions for parameters in the other structures are more straightforward to derive.