12-2009

FULLY EXPONENTIAL LAPLACE APPROXIMATION EM ALGORITHM FOR NONLINEAR MIXED EFFECTS MODELS

Meijian Zhou
University of Nebraska-Lincoln, meijian.zhou@huskers.unl.edu

FULLY EXPONENTIAL LAPLACE APPROXIMATION EM ALGORITHM FOR NONLINEAR MIXED EFFECTS MODELS

by

Meijian Zhou

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Statistics

Under the Supervision of Professor Anne M. Parkhurst

Lincoln, Nebraska

December, 2009

FULLY EXPONENTIAL LAPLACE APPROXIMATION EM ALGORITHM FOR NONLINEAR MIXED EFFECTS MODELS

Meijian Zhou, Ph.D.
University of Nebraska, 2009

Advisor: Anne M. Parkhurst

Nonlinear mixed effects models provide a flexible and powerful platform for the analysis of clustered data that arise in numerous fields, such as pharmacology, biology, agriculture, forestry, and economics. This dissertation focuses on fitting parametric nonlinear mixed effects models with single- and multi-level random effects. A new, efficient, and accurate method that gives an error of order $O(1/n^2)$, fully exponential Laplace approximation EM algorithm (FELA-EM), for obtaining restricted maximum likelihood (REML) estimates in nonlinear mixed effects models is developed. Sample codes for implementing FELA-EM algorithm in R are given. Simulation studies have been conducted to evaluate the accuracy of the new approach and compare it with the Laplace approximation as well as four different linearization methods for fitting nonlinear mixed effects models with single-level and two-crossed-level random effects. Of all approximations considered in the thesis, FELA-EM algorithm is the only one that gives unbiased or close-to-unbiased (%Bias < 1%) estimates for both the fixed effects and variance-covariance parameters. Finally, FELA-EM algorithm is applied to a real dataset to model feeding pigs' body temperature and a unified strategy for building crossed and nested nonlinear mixed effects models with treatments and covariates is provided.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Anne M. Parkhurst, for offering me a graduate research assistantship [1,2,3] and for all her inspiring guidance, full support and patience throughout this research project.

I would like to thank my supervisory committee, Dr. Kent M. Eskridge and Dr. Shunpu Zhang for giving helpful advice and perceptive comments on my dissertation, and Dr. Roger A. Eigenberg for sharing his data and helping me understand his research and experiments on heat dissipation in swine.

I would like to thank B. C. Pollard and Dr. Robert J. Collier from University of Arizona and Dr. John A. Nienaber and Dr. G. Leroy Hahn from U.S. Meat Animal Research Center for sharing their data to allow me to pursue my interest in applications of nonlinear mixed effects models.

I would also like to thank Dr. David A. Fournier for helping me develop codes for crossed and nested nonlinear mixed models in ADMB-RE and Dr. Oliver Schabenberger for providing valuable advice on fitting nonlinear mixed models with both crossed and nested random effects in the SAS macro %NLINMIX.

Special thanks to my wife Hui Shen, my daughters, Amy Zhou and Claire Zhou, my parents, Mr. Yiben Zhou and Mrs. Yuehua Zhu, my sisters, Guangqin Zhou and Xueqin Zhou and their families, for their love and support during the five years of study.

Reference

[1] U.S. Department of Agriculture, Agricultural Research Division

[2] Institute of Agriculture and Natural Resources, University of Nebraska-Lincoln

[3] Multi-State Research Project W173: Stress Factors of Farm Animals and Their Effects on Performance

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Nonlinear mixed effects models are mixed effects models in which both fixed and random effects are allowed to have a nonlinear relationship to the response variable. They are natural extensions of the nonlinear regression models for handling data from several individuals and the linear mixed effects models to the case of a nonlinear response function. Nonlinear mixed effects models provide a flexible and powerful platform for the analysis of clustered data. Since the first developments of nonlinear mixed effects models appear in Sheiner and Beal in 1980, nonlinear mixed effects models have been widely used in numerous fields, such as pharmacokinetics, biology, agriculture, environment, medicine, and economics.

1.1 Motivation

The evaluation of the log-likelihood function in nonlinear mixed effects models is a rather complex numerical issue because it involves the evaluation of a multiple integral that, in most cases, does not have a closed-form expression. Different approximation methods have been proposed to circumvent this problem. The most popular approximation methods used to estimate the parameters in nonlinear mixed effects models are linearization methods. Linearization methods use a first-order Taylor series expansion to approximate the nonlinear model function around the current estimates of the fixed effects and a choice of expansion locus for the random effects – either around zero that is the expected value of the random effects (Sheiner and Beal, 1980; Wolfinger

and Lin, 1997, zero-expansion method), or around the current estimates of the random effects (Lindstrom and Bates, 1990; Wolfinger and Lin, 1997, EBLUP-expansion method), and then maximize the likelihood corresponding to the resulting approximate linear mixed effects model. Linearization methods are computationally simple because they avoid complicated numerical integrations; however, they may produce substantial bias in parameter estimation with limited number of observations per stratum and large variability of random effects (Ge, Bickel and Rice, 2004).

This bias has motivated researchers to seek more accurate methods to improve the estimation in nonlinear mixed effects models. Laplace approximation is a popular method, which is based on using a second-order Taylor series expansion to integrate out either the random effects (Pinheiro and Bates, 1995; Vonesh, 1996), or both the random effects and the fixed effects by assuming a flat prior for the fixed effects (Wolfinger, 1993) from the marginal likelihood. While Laplace approximation methods provide more accurate estimates than linearization methods and are computationally efficient, they can also introduce a bias in parameter estimation when the number of subjects is small (Breslow and Lin, 1995; Shun and McCullagh, 1995; Kauermann, Xu, and Vaida, 2008). Other methods include Gaussian quadrature approach (Pinheiro and Bates, 1995), importance sampling approach (Pinheiro and Bates, 1995), and Markov Chain Monte Carlo technique (Wakefield, Smith, Racine-Poon, and Gelfand, 1994). These numerical integration methods are often referred to as "exact" methods in statistical literature, "exact" in the sense that they can be made as accurate as desired by taking sufficient large number of grid points or simulated samples. The numerical integration methods

work extremely well for single-level nonlinear mixed effects models with a small number of random effects (i.e., one or two), but can become computationally intensive as the number of random effects increases (Vonesh, Wang, Nie, and Majumdar, 2002).

An alternative to the direct approximate to the marginal likelihood of nonlinear mixed effects models is the use of the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977). Since the traditional EM algorithm has the same difficulty that the exact maximum likelihood approach has in fitting nonlinear mixed effects models, various simulated EM algorithms for handling nonlinear mixed effects models have received increasing interest in the statistical literature. The Monte Carlo version EM algorithms use Monte Carlo integration to approximate conditional expectations in the E-step and generate simulated samples either from the assumed distribution of the random effects (Walker, 1996) or via importance sampling from a mixture distribution that is simple in the form, easy to sample from, and efficient (Wang, 2007). The stochastic approximation version EM algorithm (Kuhn and Lavielle, 2005) replaces the E-step by a simulation step and a stochastic approximation step to obtain simulation samples from a Markov Chain Monte Carlo procedure. Like the numerical integration methods for direct approximation to the marginal likelihood of nonlinear mixed effects models, the computation of the simulated EM algorithms can also be challenging as the number of random effects increases.

The purpose of this dissertation is to develop a new, efficient, and accurate method, fully exponential Laplace approximation EM algorithm (FELA-EM), for obtaining restricted

maximum likelihood (REML) estimates in parametric nonlinear mixed effects models with single- and multi-level random effects. The proposed FELA-EM algorithm gives more accurate estimates, an error of order $O(1/n^2)$, than those from Laplace approximation, an error of order $O(1/n)$, while preserving the numerical simplicity of Laplace approximation and thus, the proposed FELA-EM algorithm is computationally much simpler than numerical integration methods and simulated EM algorithms.

1.2 Summary of the remaining chapters

In Chapter 2, we review the literature of four kinds of nonlinear mixed effects models (parametric modeling, nonparametric modeling, semi-parametric modeling, and Bayesian modeling) and their corresponding parameter estimation methods. A review of software packages available for fitting mixed effects models is also given.

In Chapter 3, we propose the EM algorithm using the fully exponential Laplace method to approximate the conditional expectations of the complete data sufficient statistics in the E-step for obtaining restricted maximum likelihood estimates in single-level nonlinear mixed effects models. We compare the FELA-EM algorithm with the Laplace approximation and four different linearization methods using simulation results to evaluate the accuracy of the new approach. These comparisons provide a useful foundation for the relative potential strengths and weaknesses of the considered estimation methods for fitting nonlinear mixed effects models.

In Chapter 4, the FELA-EM algorithm for single-level nonlinear mixed effects models is generalized to fit multi-level nonlinear mixed effects models with two crossed random effects, to which none of the numerical integration methods and simulated EM algorithms are currently directly applicable.

Chapter 5 contains an application of the proposed FELA-EM algorithm to a real dataset, where a multilevel nonlinear mixed effects model with both crossed and nested random effects applied in a replicated Latin square design is developed to model feeding pigs' body temperature in conjunction with three thermal environmental treatments, the amount of feed intake, and the duration of the meal. Random effects associated with three classification factors are introduced into a modified first-order compartment model. The within-event correlation is described by an AR(1) model.

1.3 References

Breslow, N.E., Lin, X., 1995. Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika* 82, 81-91.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society*, Series B 39, 1-38.

Ge, Z., Bickel, J.P., Rice, A.J., 2004. An approximate likelihood approach to nonlinear mixed effects models via spline approximation. *Computational Statistics & Data Analysis* 46, 747-776.

Kauermann, G., Xu, R., Vaida, F., 2008. Stacked Laplace-EM algorithm for duration models with time-varying and random effects. *Computational Statistics & Data Analysis* 52, 2514-2528.

Kuhn, E., Lavielle, M., 2005. Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics & Data Analysis* 49, 1020-1038.

Lindstrom, M.J., Bates, D.M., 1990. Nonlinear mixed effects models for repeated measures data. *Biometrics* 46, 673-687.

Pinheiro, J.C., Bates, D.M., 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4, 12-35.

Sheiner, L.B., Beal, S.L., 1980. Evaluation of methods for estimating population pharmokinetics. I. Michaelis-Menton Model: Routine clinical pharacokinetics data. *Journal of Pharmacokinetics and Biopharmaceutics* 8, 673-687.

Shun, Z., McCullagh, P., 1995. Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society*, Series B 57, 749-760.

Vonesh, E.F., 1996. A note on the use of Laplace's approximation for nonlinear mixed-effects models. *Biometrika* 83, 447-452.

Vonesh, E.F., Wang, H., Nie, L., Majumdar, D., 2002. Conditional second-order generalized estimating equations for generalized linear and nonlinear mixed-effects models. *Journal of the American Statistical Association* 97, 271-283.

Wakefield, J.C., Smith, A.F.M., Racine-Poon, A., Gelfand, A.E., 1994. Bayesian analysis of linear and nonlinear population models by using the Gibbs sampler. *Applied Statistics* 43, 201-221.

Walker, S., 1996. An EM algorithm for nonlinear random effects models. *Biometrics* 52, 934-944.

Wang, J., 2007. EM algorithms for nonlinear mixed effects models. *Computational Statistics & Data Analysis* 51, 3244-3256.

Wolfinger, R., 1993. Laplace's approximation for nonlinear mixed models. *Biometrika* 80, 791-795.

Wolfinger, R.D., Lin, X., 1997. Two Taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis* 25, 465-490.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Mixed effects models are often used to handle correlation that arises in longitudinal or other clustered data (Crowder and Hand, 1990; Verbeke and Molenberghs, 2000; Littell, Milliken, Stroup, Wolfinger, and Schabenberger, 2006). Longitudinal data can be loosely defined as data in the form of repeated measurements on the same unit over time or over another set of conditions, like repeated measurement of physiological effect on the same subject in response to different doses of a drug in pharmacodynamic studies. The unit in which the observations are measured repeatedly is also referred to as the individual or cluster and may correspond to diverse entities such as humans, animals, plants, laboratories, or experiments. Longitudinal data appear, frequently, both in observational studies which are longitudinal in nature, and in experimental studies incorporating repeated measures designs and include a broad range of research areas, such as forestry, agriculture, animal science, life sciences, pharmacokinetics, medical and public health research. Examples occur in many fields:

- In forestry, a measure of growth may be taken on the same tree monthly over several years and trees are associated with different site preparation treatments or soil types.
- In animal science, a measure of the body temperature may be taken at hourly intervals on the same animal and animals are assigned to different thermal environments.

- In pharmacokinetics, a measure of the drug concentration may be taken at several times on the same rat following administration and rats are initially given different amounts of oral dose.

The scientific questions often involve not only how the mean response differs among treatments, but also how the mean response changes over time and whether or not that change depends on treatment effects.

Observations on the same unit usually cannot be considered independent and mixed effects models provide a convenient way for modeling such dependence. In the mixed effects model, the response is assumed to be a function of fixed effects, non-observable individual specific random effects, and an error term. Observations within the same unit share common random effects and are therefore statistically dependent.

A mixed effects model contains two types of parameters: fixed effects and variance-covariance components. In many practical applications, estimates of the random effects, which are associated with individual units randomly drawn from a population, are also of interest.

Different estimation methods have been studied for mixed effects models. While methods (maximum likelihood and restricted maximum likelihood) for solving linear mixed effects models are available in many software packages and described in many references (Harville, 1977; Longford, 1993; Pinheiro and Bates, 2000; SAS Institute,

2004), methods for nonlinear mixed effects models are still being explored (Davidian and Giltinan, 1995; Vonesh and Chinchilli, 1997; Davidian and Giltinan, 2003).

2.2 Linear mixed effects models

Linear mixed effects models are mixed effects models in which both the fixed and the random effects enter linearly into the model function. Laird and Ware (1982) have presented a general form of such models,

$$\begin{aligned} &\boldsymbol{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon}_i, \qquad i = 1,2,...,M, \\ &\boldsymbol{b}_i \sim N(\mathbf{0}, \mathbf{D}), \qquad \boldsymbol{\varepsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Psi}), \end{aligned} \tag{2.2.1}$$

where $\boldsymbol{y}_i$ is an $(n_i \times 1)$ vector of responses for the ith unit, $\boldsymbol{\beta}$ is an $p \times 1$ vector of fixed effects, $\boldsymbol{b}_i$ is an $q \times 1$ vector of random effects distributed as normal with mean $\mathbf{0}$ and variance-covariance matrix $\mathbf{D}$, $\mathbf{X}_i$ is an $n_i \times p$ design matrix corresponding to fixed effects, $\mathbf{Z}_i$ is an $n_i \times q$ design matrix corresponding to random effects, and $\boldsymbol{\varepsilon}_i$ is an $n_i \times 1$ vector of within unit errors and is normally distributed with mean $\mathbf{0}$ and variance-covariance matrix $\boldsymbol{\Psi}$. It is further assumed that the random effects $\boldsymbol{b}_i$ and the within unit errors $\boldsymbol{\varepsilon}_i$ are independent for different units and independent of each other for the same unit.

Several methods have been proposed to estimate parameters in the linear mixed effects models and the most common ones are likelihood-based methods: maximum likelihood

(ML) and restricted maximum likelihood (REML). To find (RE)ML estimates, we need to maximize the objective function associated with ML or REML over all unknown parameters. The optimization is usually conducted using the Expectation-Maximization (EM) algorithm (Dempster, Laird and Rubin, 1977) or Newton-Raphson methods (Thisted, 1988). There are no closed form expressions for the distribution of (RE)ML estimates and inference (hypothesis tests and confidence intervals) are generally based on asymptotic normality.

2.3 Nonlinear regression models

Nonlinear regression is a powerful tool for analyzing scientific data. Many real-life phenomena can be described by a nonlinear regression function, such as in pharmacology, physiology, biology, and etc. Usually, the model parameters are easier to interpret compared to those from linear models because the parameters in nonlinear models generally have a natural physical interpretation. Before the age of computers, nonlinear regression was not readily available to most scientists. Instead, scientists transformed their data to make a linear relationship, and then analyzed the transformed data with linear regression. These methods are outdated and should not be used to analyze data since the linear transformation always distorts the experimental error. Estimation for a nonlinear regression model is an iterative process. The general steps for fitting a nonlinear regression model are: (1) Give an initial value for each parameter in the model; (2) Generate the curve defined by the initial values and calculate the residual sum of squares; (3) Adjust the value for each parameter to make the curve come closer to the data points. There are a number of different algorithms for adjusting the value for each

parameter, such as steepest-descent or gradient method, Newton method, Gauss-Newton method, Marquardt method, etc. All these methods use derivatives (or approximations to derivatives) of the residual sum of squares with respect to the parameters to search for the parameter estimates that produce the smallest residual sum of squares; (4) Use the adjusted estimates as new starting values. Repeat steps (2), (3), and (4) until a termination criterion is satisfied.

2.4 Nonlinear mixed effects models

Nonlinear mixed effects models are mixed effects models in which some of the fixed and/or random effects enter nonlinearly to the model function. Nonlinear mixed effects models may be regarded as an extension of 1) the nonlinear regression models to fit data from several individuals and 2) the linear mixed effects models to the case of a nonlinear response function. Nonlinear mixed effects models are a popular platform for analyzing clustered data when interests focuses on individual-specific characteristics. They first received widespread attention in the statistical literature in the late 1980s, and a number of new computational methods for these models were developed in the 1990s. Currently, nonlinear mixed effects models have been widely used in numerous fields, such as biology, agriculture, environment, medicine, and economics, and several different general-purpose software packages are also available.

The general form of nonlinear mixed effects models is as follows:

$$\boldsymbol{y} = f(\boldsymbol{\beta}, \boldsymbol{b}) + \boldsymbol{\varepsilon} \tag{2.4.1}$$

where $\boldsymbol{y}$ is the response vector, f is a general nonlinear function, $\boldsymbol{\beta}$ is the vector of fixed effects, $\boldsymbol{b}$ is the vector of random effects, and $\boldsymbol{\varepsilon}$ is the error vector. It is further assumed that the random effects $\boldsymbol{b}$ have a density and the errors $\boldsymbol{\varepsilon}$ are normally distributed with mean $\mathbf{0}$ and variance-covariance matrix $\boldsymbol{\Psi}$, with $\boldsymbol{b}$ independent of $\boldsymbol{\varepsilon}$.

The first developments of nonlinear mixed effects models go back to Sheiner and Beal (1980). Since then, a number of statistical approaches to nonlinear mixed effects modeling have been discussed in the literature and applied in data analyses. Basically, they can be classified into four categories: parametric modeling, nonparametric modeling, semi-parametric modeling, and Bayesian modeling.

2.5 Four categories of nonlinear mixed effects models

2.5.1 Parametric nonlinear mixed effects models

For parametric nonlinear mixed effects models, we have a model similar to (2.4.1) with a further assumption that the random effects $\boldsymbol{b}$ have a normal distribution. A major complication of parameter estimation in the parametric nonlinear mixed effects model is the likelihood function, which is based on the marginal distribution of $\boldsymbol{y}$, does not usually have a closed form solution. Different methods have been presented for estimating the parameters in the parametric nonlinear mixed effects model and there is an ongoing debate in the literature about which is the most adequate method. Most of the estimation methods can be divided into three categories: linearization methods (Sheiner and Beal, 1980; Lindstrom and Bates, 1990; Wolfinger and Lin, 1997), integral

approximation methods (Wolfinger, 1993; Pinheiro and Bates, 1995; Vonesh, 1996), and EM algorithms (Walker, 1996; Kuhn and Lavielle, 2005; Wang, 2007). The linearization methods approximate the nonlinear mixed effects model by a first-order Taylor series expansion to arrive at a pseudo-model that is typically of the linear mixed effects model form. Integral approximation methods use Laplace approximation, Gaussian quadrature, or importance sampling to calculate the marginal distribution of $\boldsymbol{y}$ and then maximize the likelihood directly. As an alternative to the direct approximate to the marginal likelihood, EM algorithms approximate the conditional expectation of the log likelihood in the E step and then maximize the expected log likelihood to obtain the estimates in the next E step. The linearization methods are by far the most popular approaches due to their numerically simplicity. More details are discussed in Section 2.6.

2.5.2 Nonparametric nonlinear mixed effects models

For nonparametric nonlinear mixed effects models, a maximum likelihood method was proposed by Mallet, Mentre, Steimer and Lokiek (1988). The difference between their model and a parametric model is that they make no assumptions about the distribution of the random effects, except that it is a probability measure. The conditional distribution of the response $\boldsymbol{y}$ given the random effects is assumed to be known. The objective of the estimation procedure is to get the probability distribution of the cluster-specific effects, $\boldsymbol{b}$, that maximizes the likelihood of the data. Mallet (1986) proved that the maximum likelihood solution is a discrete distribution with the number of discontinuity points less or equal to the number of clusters in the sample.

2.5.3 Semi-parametric nonlinear mixed effects models

For semi-parametric nonlinear mixed effects models, a maximum likelihood estimation method was proposed by Davidian and Gallant (1992). The difference between their model and a parametric model is that they allow the distribution of the random effects to be free to vary within a class of smooth densities, $\boldsymbol{H}$, defined in Gallant and Nychka (1987). Densities in $\boldsymbol{H}$ may be skewed, multi-modal, and fat-tailed or thin-tailed relative to the k-variate normal density. Class $\boldsymbol{H}$ also contains the normal density. However, densities in $\boldsymbol{H}$ may not exhibit unusual behavior such as kinks, jumps, or oscillation. A density from $\boldsymbol{H}$ can be expressed as an infinite linear combination of normal densities. In the likelihood calculations the summation is truncated to a finite number of terms and numerical integration is carried out using Gauss-Hermite quadrature. This semi-parametric approach is implemented in the Nlmix software, available through StatLib (Davidian and Gallant, 1992).

2.5.4 Bayesian approach to nonlinear mixed effects models

A Bayesian approach to nonlinear mixed effects models is proposed by Bennett and Wakefield (1993), Wakefield (1993), and Wakefield, Smith, Racine-Poon, and Gelfand (1994). They use a three-stage model and Markov Chain Monte Carlo (MCMC). In the first stage, they specify the conditional density of the observation $\boldsymbol{y}$ given the random effects $\boldsymbol{b}$ and the fixed effects $\boldsymbol{\beta}$. In the second stage, they specify the density of the random effects $\boldsymbol{b}$ given $\boldsymbol{\beta}$, and in the third stage, a prior density for $\boldsymbol{\beta}$. The posterior density of the random effects can be obtained using Markov chain Monte Carlo methods and then any desired feature of the posterior density, such as the mode, moments,

probabilities and credible intervals can be approximated. MCMC techniques have been implemented in WinBUGS (Spiegelhalter, Thomas, Best and Lunn 2003), which is an all-purpose software package for Bayesian analysis. The most popular MCMC method is the Gibbs Sampler. For a detailed account of the Gibbs Sampler and general MCMC, see Gilks, Richardson and Spiegelhalter (1996). MCMC has a great potential in its ability to handle mixed effects models with high dimensional random effects given currently available computing resources. However, because of difficulties of assessing convergence to stationarity and the error in estimates, Evans and Swartz (1995) comment that Markov chain methods are recommended only when there are no other adequate alternatives.

2.6 Estimation methods for parametric nonlinear mixed effects models

There are different formulations of nonlinear mixed effects models available in the literature. In this dissertation, we focus on the parametric nonlinear mixed effects model with normally distributed random-effects and error terms, which is given by

$$y_{ij} = f(\boldsymbol{\beta}, \boldsymbol{b}_i) + \varepsilon_{ij}, \quad i = 1,...,M, \quad j = 1,...,n_i, \tag{2.6.1}$$

where y_{ij} is the jth observation on the ith subject, f is a nonlinear function, $\boldsymbol{\beta}$ is a p-dimensional vector of fixed effects, $\boldsymbol{b}_i$ is a q-dimensional random effects vector associated with the ith subject (not varying with j) and assumed i.i.d. normal with mean $\mathbf{0}$ and variance-covariance matrix $\mathbf{D}$, ε_{ij} is the error and assumed i.i.d. $N(0, \sigma^2)$, M is

the number of subjects, and n_i is the number observations on the ith subject. It is further assumed that b_i and ε_{ij} are independent. As mentioned in Section 2.5.1, there are three main approaches for parametric nonlinear mixed effects models: linearization methods, integral approximation methods, and EM algorithms, and all of them are approximation methods since the likelihood function of the parametric nonlinear mixed effects model does not generally have a closed form solution.

2.6.1 Linearization methods

The basic idea of linearization is as follows: 1) Take a first-order Taylor series expansion of the model around some values of the random effects $\boldsymbol{b}$ and the current estimates of the fixed effects $\boldsymbol{\beta}$, which yields a pseudo-model that is of the linear mixed effects model form; 2) Fit this pseudo-model; 3) Update the values of the random effects and the estimates of the fixed effects; 4) Repeat the process until a convergence criterion is met. The first-order Taylor series of the model function $f(\boldsymbol{\beta}, \boldsymbol{b}_i)$ in the model (2.6.1) around $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}_i$ is

$$f(\boldsymbol{\beta}, \boldsymbol{b}_i) \approx f(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i) + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right) + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{b}_i}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \left(\boldsymbol{b}_i - \hat{\boldsymbol{b}}_i\right). \tag{2.6.2}$$

Substituting for $f(\boldsymbol{\beta}, \boldsymbol{b}_i)$ in (2.6.1) with (2.6.2), we obtain

$$y_{ij} \approx f(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i) + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right) + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{b}_i}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \left(\boldsymbol{b}_i - \hat{\boldsymbol{b}}_i\right) + \varepsilon_{ij}. \tag{2.6.3}$$

Rearranging terms in (2.6.3) produces the following approximate model that is of the linear mixed effects model form

$$
\begin{aligned}
y_{ij}' &= y_{ij} - f(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i) + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \hat{\boldsymbol{\beta}} + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{b}_i}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \hat{\boldsymbol{b}}_i \\
&\approx \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \boldsymbol{\beta} + \left.\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{b}_i)}{\partial \boldsymbol{b}_i}\right|_{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}_i} \boldsymbol{b}_i + \varepsilon_{ij} .
\end{aligned}
\tag{2.6.4}
$$

Linearization methods are natural extensions of the linearization algorithms for classical nonlinear regression. The advantages of linearization methods are: 1) it is computationally simple; 2) multiple levels of nested or crossed random effects can be accommodated; 3) it is implemented in popular software packages such as NLINMIX macro in SAS and NLME in both S-PLUS and R. However, despite its popularity, the drawbacks of linearization methods are: 1) it may produce substantial bias in parameter estimation with limited number of observations per unit and large variability of random effects (Ge, Bickel and Rice, 2004); 2) it maximizes the likelihood of some pseudo-data, not the original data.

Linearization methods differ in the expansion locus of the random effects. Beal and Sheiner (1982) propose a first-order method, which approximates the likelihood by taking a first-order Taylor series expansion of the model function around the mean of the random effects (Notice: the mean of any random effects is simply 0). They have

implemented their method in the software package NONMEM (Beal and Sheiner, 1992), which is widely used in pharmacokinetics. It is also available in SAS proc nlmixed (SAS Institute, 2004) via the method=firo option. Lindstrom and Bates (1990) suggest a more accurate approximation compared to the first-order method, which takes a first-order Taylor series expansion of the model function around the current estimated best linear unbiased predictor (BLUP) of the random effects. Their method is implemented in another popular software package, called NLME (Pinheiro and Bates 2000), available in both S-PLUS (Insightful Corporation 2007) and R (R development core team 2009). The function NLME can only be used to solve nonlinear mixed effects model with nested random effects. Wolfinger and Lin (1997) use a different algorithm to implement both Beal and Sheiner's and Lindstrom and Bates' linearization methods in the SAS macro NLINMIX. NLINMIX can be used to solve nonlinear mixed effects model with both crossed and nested random effects.

2.6.2 Integral approximation methods

The basic idea of integral approximation is as follows: 1) Approximate the marginal likelihood of the response $\boldsymbol{y}$ using a technologically available numerical integration routine; 2) Maximize the approximated likelihood numerically. Integral approximations are generally computationally more demanding than linearization methods. However, integral approximations usually maximize the likelihood of the original data and they can generate more consistent and accurate estimates in parameter estimation compared to linearization methods (Schabenberger and Pierce 2001). Therefore, it is usually a good idea to use linearization methods to provide starting values for the more accurate integral

approximation methods. Different methods based on integral approximation have been proposed to fit nonlinear mixed effects models. They include Laplace approximation, Importance Sampling, and Gaussian quadrature methods.

The Laplace approximation is a method for approximating integrals using local information about the integrand at its maximum. Therefore, it is most useful when the integrand is highly concentrated about its maximizing value. The Laplace approximation has been widely used in Bayesian analyses to compute posterior means and variances of parametric functions and it is also useful for approximating the likelihood in nonlinear mixed effects models when the integrals in the likelihood do not have closed form solutions (Naylor and Smith, 1982; Tierney and Kadane, 1986; Leonard, Hsu and Tsui, 1989; Tierney, Kass and Kadane, 1989). In both cases, the Laplace approximation converges to the correct value of the integral as the sample size goes to infinity. Given a one-dimensional integral of a positive function $p(b)$ and denoted the log of the positive function $p(b)$ as $l(b)$, the Laplace approximation in its simplest form is

$$\int p(b)db \approx \sqrt{2\pi}\exp(l(\hat{b}))(-l''(\hat{b}))^{-\frac{1}{2}}, \tag{2.6.5}$$

where $\hat{b}$ maximizes $l(b)$. Note that this form of the approximation only applies to positive integrands. The idea for this approximation comes from a second order Taylor series expansion of $l(b)$ about $b = \hat{b}$. That is, for b near $\hat{b}$,

$$l(b) \approx l(\hat{b}) + l'(\hat{b})(b-\hat{b}) + \frac{1}{2} l''(\hat{b})(b-\hat{b})^2,$$

and $l'(\hat{b}) = 0$ since $\hat{b}$ maximizes $l(b)$. Thus

$$\begin{aligned}
\int p(b)db &= \int \exp(l(b))db \\
&\approx \int \exp(l(\hat{b}) + \frac{1}{2} l''(\hat{b})(b-\hat{b})^2)db \\
&= \exp(l(\hat{b}))\int \exp(\frac{1}{2} l''(\hat{b})(b-\hat{b})^2)db \\
&= \exp(l(\hat{b}))\sqrt{\frac{2\pi}{-l''(\hat{b})}}\int \sqrt{\frac{-l''(\hat{b})}{2\pi}} \exp(-(-l''(\hat{b})/2)(b-\hat{b})^2)db \\
&= \sqrt{2\pi}\exp(l(\hat{b}))(-l''(\hat{b}))^{-\frac{1}{2}},
\end{aligned}$$

since $\sqrt{\frac{-l''(\hat{b})}{2\pi}} \exp(-(-l''(\hat{b})/2)(b-\hat{b})^2)$ is the density function of a normal distribution with mean $\hat{b}$ and variance $-1/l''(\hat{b})$ and then $\int \sqrt{\frac{-l''(\hat{b})}{2\pi}} \exp(-(-l''(\hat{b})/2)(b-\hat{b})^2)db = 1.$

Given q-dimensional integrals of a positive function $p(b)$ and the log of the integrand $l(b)$, the Laplace approximation with similar results is

$$\int p(b)db \approx (2\pi)^{\frac{q}{2}} \exp(l(\hat{b}))\left|-l''(\hat{b})\right|^{-\frac{1}{2}}, \qquad (2.6.6)$$

where $\hat{b}$ maximizes $l(b)$ and $\left|-l''(\hat{b})\right|$ denotes the determinant of the matrix $(-l''(\hat{b}))$. Equation (2.6.6) differs from the univariate case (2.6.5) in that the second derivative of $l(b)$ has been replaced by the determinant of the matrix of second-order derivatives, and the power of 2π is multiplied by the dimension q to give a power of $q/2$. Laplace approximation to nonlinear mixed effects models has different variations. The ML version proposed by Pinheiro and Bates (1995) treats the marginal likelihood of the nonlinear mixed effects model as an integral with respect to the random effects only while the REML version presented by Wolfinger (1993) integrates out both the random effects and the fixed effects by assuming a flat prior for the fixed effects from the marginal likelihood of nonlinear mixed effects models. While Laplace approximation provides more accurate estimates than linearization methods and is computationally efficient, the ML version Laplace approximation procedure can introduce a bias for mixed effects models when the number of subjects is small (Breslow and Lin, 1995; Shun and McCullagh, 1995; Kauermann, Xu, and Vaida, 2008). The performance of the REML version Laplace approximation still needs further investigated. The ML version Laplace approximation is implemented in the nlmixed procedure of SAS (SAS Institute, 2004) for fitting single-level nonlinear mixed effects models. Bates, Maechler, and Dai (2008) extend the ML version of Laplace approximation for single-level nonlinear mixed effects models to multilevel nonlinear mixed effects models with both crossed and nested random effects and implement their approach in the new R function NLMER.

Importance sampling is a common method to approximate an integral numerically. It takes advantage of the fact that any integral can be thought of as an expectation function. For an arbitrary multiple integral, we can always represent it by

$$I = \int f(x)dx = \int \frac{f(x)}{p_X(x)} p_X(x)dx = E(\frac{f(x)}{p_X(x)}), \quad x \in R^n, \tag{2.6.7}$$

where X is any random variable with p.d.f. $p_X(x)$ and $E(\cdot)$ represents the expectation. We then take a sample $X_1,\ldots,X_n$ from the p.d.f. of $p_X(x)$ and approximate the integral (2.6.7) by the sample mean $\hat{I} = \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{p_X(x_i)}$. Importance Sampling provides a simple and efficient way of performing Monte Carlo integration. The critical step for the success of this method is the choice of an importance distribution from which the sample is drawn and importance weights are calculated. For more details about Importance sampling, see Pinheiro and Bates (1995).

Gaussian quadrature method makes use of the Gauss-Hermite quadrature rule to approximate an integral. It essentially replaces the integral with a fixed set of spaced intervals and associated weights. In the univariate case, the integral $\int h(x)dx$ can be approximated by

$$\int h(x)dx = \int f(x)w(x)dx \approx \sum_{i=1}^{N_{GQ}} w_i f(x_i) \tag{2.6.8}$$

where $h(x) = f(x)w(x)$, $w(x)$ is the weight function, w_i are weights, x_i are abscissas, and N_{GQ} is the number of the quadrature points. If the intervals are selected around the conditional mode of the random effects, the resulting quadrature approximation is called adaptive Gaussian quadrature approximation. For more details about Gaussian quadrature approximation, see Abramowitz and Stegun (1964), Golub and Welsch (1969), Golub (1973), Pinheiro and Bates (1995).

Numerical integration methods including Gaussian quadrature, importance sampling, and other Monte Carlo (MC) integration methods work extremely well for nonlinear mixed effects models with a small number of random effects (e.g., one or two). However, they become computationally intensive as the number of random effects increases (Vonesh, Wang, Nie, and Majumdar, 2002).

2.6.3 EM algorithms

The EM algorithm introduced by Dempster, Laird, and Rubin (1977) is an iterative method that alternates between performing an expectation step (E-step) and a maximization step (M-step). Its simplicity and stability have made it a popular approach for finding maximum likelihood estimates of parameters in statistical models that depend on missing data or unobserved variables. Let $\boldsymbol{y}_{obs}$, $\boldsymbol{y}_{mis}$, and $\boldsymbol{\theta}$ represent the observed data, missing data, and the vector of parameters to be estimated, respectively. On the $(k+1)$th iteration, the E-step generally computes the expectation of the complete data log-likelihood $\ell(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis}; \boldsymbol{\theta})$ with respect to the conditional distribution of the missing data $\boldsymbol{y}_{mis}$ given the observed data $\boldsymbol{y}_{obs}$ under the current estimates of the parameters $\boldsymbol{\theta}^{(k)}$,

$$Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)}) = E[\ell(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis}; \boldsymbol{\theta})\big|\boldsymbol{y}_{obs}; \boldsymbol{\theta}^{(k)}].$$

The M-step is then to find $\boldsymbol{\theta}^{(k+1)}$ to maximize $Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)})$ for all $\boldsymbol{\theta}$ in the parameter space $\boldsymbol{\Omega}$. The two steps are repeated until convergence.

For nonlinear mixed effects models, similar to the evaluation of the log-likelihood function, the assessment of the expectation $Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)})$ also involves the evaluation of a multiple integral that, in most cases, does not have a closed-form expression. Various simulated EM algorithms for nonlinear mixed models have been presented to approximate the E-step. Walker (1996) suggests using Monte Carlo integration to handle conditional expectations in the E-step and he also shows that the M-step is analytically tractable. The simulated samples used in Walker (1996) are from the assumed distribution of the random effects. Wang (2007) presents a Monte Carlo EM (MCEM) algorithm that uses samples obtained from an easy-to-simulate and efficient importance distribution so that the computational intensity and complexity can be somewhat reduced. The stochastic approximation version EM (SAEM) algorithm proposed by Kuhn and Lavielle (2005) decomposes the E-step into a simulation step and a stochastic approximation step and samples of missing data are either simulated under the conditional distribution $p(\boldsymbol{y}_{mis}\big|\boldsymbol{y}_{obs}; \boldsymbol{\theta}^{(k)})$ or obtained from Markov Chain Monte Carlo (MCMC) procedure. The SAEM algorithm requires the simulation of only one realization of the missing data for each iteration and, thus, substantially reduces the

computation time compared to MCEM. Like that of the numerical integration methods for direct approximation to the marginal likelihood of nonlinear mixed effects models, the computation of the simulated EM algorithms can also be intensive as the number of random effects increases.

Research on parameter estimation methods for the nonlinear mixed model is still in progress. There are questions on how to implement the methods, how well the methods work, what are the asymptotic and finite-sample properties. For example, Vaida, Fitzgerald, and DeGruttola (2007) discuss a hybrid Monte Carlo and numerical integration EM algorithm for computing the maximum likelihood estimates for linear and nonlinear mixed models with censored data. Noh and Lee (2008) propose the use of Lee and Nelder (1996) hierarchical-likelihood approach for the analysis of nonlinear mixed effects model and show that it gives statistically and computationally efficient estimates. Panhard and Samson (2009) use an extension of the stochastic approximation version of EM (SAEM) algorithm for obtaining maximum likelihood estimates of multilevel nonlinear mixed effects models with two nested random effects and show that their approach can achieve gains in accuracy over the linearization methods.

2.7 Software review

Software packages offering the capacity to fit nonlinear mixed effects models are available from a few sources such as NONMEM, SAS, S-PLUS/R, and random effects modeling in AD Model Builder (ADMB-RE). The first software package developed to fit nonlinear mixed effects models is NONMEM (Beal and Sheiner, 1992), which approximates maximum likelihood estimation based on the first-order and conditional first-order methods. NONMEM has been widely used in the area of pharmacokinetic and pharmacodynamic analysis and it currently does not handle crossed random effects.

With Release 8.0 and the later version, the NLMIXED procedure in SAS is available to fit nonlinear mixed effects models with Gaussian errors (The MIXED/NLMIXED Procedure, 1999). The NLMIXED procedure approximates the integral for a nonlinear mixed effects model using Laplace approximation, importance sampling, and adaptive Gauss quadrature methods together with Beal and Sheiner's first-order method. So far, the NLMIXED procedure allows only one random statement, which limits nonlinear models to data with a single classification factor, that is, the NLMIXED procedure can only handle nonlinear mixed models without nested and crossed random effects.

A SAS macro NLINMIX fits nonlinear mixed effects models with Gaussian outcomes based on linearization methods (Wolfinger and Lin, 1997). The NLINMIX macro can handle nonlinear mixed models with both crossed and nested random effects.

The package NLME, written by Pinheiro and Bates, is very powerful for fitting nonlinear mixed effects models with nested random effects based on Lindstrom and Bates' linearization method. NLME is available in both S-PLUS and R. A general description of the capabilities of NLME is given by Pinheiro and Bates (2000).

A latest developed R function NLMER fits nonlinear mixed effects models by the ML version Laplace approximation (Bates, Maechler, and Dai, 2008). NLMER can handle nonlinear mixed effects models with both crossed and nested random effects.

The software package ADMB-RE (Skaug and Fournier, 2006) can handle nonlinear mixed models with both crossed and nested random effects. ADMB-RE evaluates the marginal likelihood by either the Laplace approximation or importance sampling and calculates exact derivatives using Automatic Differentiation (AD). Sampling from the Bayesian posterior in ADMB-RE uses MCMC (Metropolis-Hastings algorithm). Automatic Differentiation refers to a collection of techniques that exploit the chain rule of calculus to automatically evaluate derivatives of functions defined in computer programs. To use ADMB-RE, users need to formulate the likelihood function in a template file using a C++ like language and then turn the template file into an executable program using a C++ compiler. As noted by the authors, the program runs very slowly when handling random effects models. The reason is that integration of the likelihood, the way ADMB-RE deals with random effects, is more computationally intensive than optimization.

There are many other popular software packages which can only handle linear mixed and/or generalized linear models, but not nonlinear mixed ones. A major statistical software package, SPSS (Statistical Package for the Social Sciences) is among the most widely known and used programs for statistical analysis in social science (Landau and Everitt, 2004). However, multilevel modeling in SPSS has definite limitations. First, it cannot fit nonlinear mixed models. Second, for fitting linear mixed models, it can not specify the covariance matrix at the lowest level. In addition, the restriction to normal response models means that it cannot handle multilevel logistic regression and multilevel Poisson regression models.

STATA is an integrated statistical package for Windows and other platforms such as UNIX, Macintosh and LINUX (Rabe-Hesketh and Everitt, 2004). This package is becoming more widely used in the statistical community. STATA Corporation has developed a set of commands for longitudinal data under the *xt* prefix. However, as the commands were not developed to handle hierarchical data, only variance component models were available in the STATA 7.0 core package. One user-defined command, *gllamm*, extends STATA capacity to fit hierarchical models, including generalized linear latent and mixed models.

SYSTAT is one of the products from SYSTAT Software Inc. (Hedeker, Marcantonio and Pechnyo, 2000). It is a statistics and graphics package for technical professionals in the areas of Data Analysis and Modeling. In the past twenty years, it has been used in the fields of Life Sciences, Bio-medical, Environmental Sciences, Automobile,

Pharmaceuticals and etc. With Release 10 of SYSTAT the mixed regression tool has become available for random effects analysis of hierarchical data. However, SYSTAT cannot handle nonlinear and nonlinear mixed models.

The Mixed-Up Suite is a family of standalone programs that fits 2-level mixed effects models (Hedeker and Gibbons, 1996a and b). The first family member MIXOR that fits random-effects probit and logistic model for ordinal outcomes came out in 1993, followed by MIXREG, MIXNO, MIXPREG and MIXGSUR. The Mixed-Up Suite can be used to fit mixed-effects linear regression, mixed-effects logistic regression for nominal or ordinal outcomes, mixed-effects probit regression for ordinal outcomes, mixed-effects Poisson regression, and mixed-effects grouped-time survival analysis. Thus, it cannot handle nonlinear mixed models. As a free package with friendly interface and good documentation to cover standard two-level random effect models for Normal, categorical and count outcomes, the Mixed-Up Suite is a very good tool for researchers and newcomers to multilevel modeling.

HLM (Hierarchical Linear Models) is a stand-alone software package handling mixed effects models and has been a popular program used by professional researchers and research students worldwide in areas of education and social science research as well as public health research (Raudenbush, Bryk, Cheong and Congdon, 2001). With the version 5.04 for Windows, HLM allows for the analysis of linear and generalized linear mixed models with two or three levels of NESTING.

MLwiN has been created by the Centre for Multilevel Modeling team based at the Institute of Education together with various colleagues in other centers (Browne, 2003; Rasbash, Steele and Browne, 2003; Yang and Goldstein, 2003). The package can be used to fit linear and generalized linear mixed models with both nested and crossed random effects but it does not handle either nonlinear or nonlinear mixed models.

GenStat was first developed in the 1960s at Rothamsted Experimental Station for use in design and analysis of agricultural experiments and has been in continuous development since (VSN International Ltd., 2006). The major user group of GenStat is statisticians and scientists working in biological research. Although most areas of statistical application are covered, GenStat's particular strengths are in its ANOVA algorithm, which analyses balanced multi-level data, and the efficient REML algorithm which analyses multi-level data, allowing for correlated errors at any level of the data. Currently, GenStat supports the analysis of linear and generalized linear mixed models.

EGRET was originally developed at the School of Public Health of University of Washington USA (Mauritsen, R.H., 1984) and it is widely used by Epidemiologists and Biostatisticians (Cytel Software Corporation, 2000). Being dedicated to binomial and count data as well as survival data, EGRET is unable to fit Normal response models. EGRET concentrates on models for categorical data collected from Epidemiology and Biomedical studies and can be used to fit generalized linear models with and without random effects and survival models.

2.8 References

Abramowitz, M., Stegun, I.A., 1964. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. New York: Dover.

Bates, D., Maechler, M., Dai, B., 2008. Lme4: Linear mixed-effects models using S4 classes. R package version 0.999375-22. http://lme4.r-forge.r-project.org

Beal, S.L., Sheiner, L.B., 1982. Estimating population kinetics. *CRC Critical Reviews in Biomedical Engineering* 8: 195-222.

Beal, S.L., Sheiner, L.B., 1992. NONMEM User's Guides. NONMEM Project Group, University of California, San Francisco.

Bennett, J.E., Wakefield, J.C., 1993. Markov Chain Monte Carlo for Nonlinear Hierarchical Models. Technical Report TR-93-11, Statistics Section, Imperial College.

Breslow, N.E., Lin, X., 1995. Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika* 82, 81-91.

Browne, W., 2003. MCMC estimation in MLwiN. Institute of Education, London.

Crowder, M.J., Hand, D.J., 1990. Analysis of Repeated Measures. London: Chapman & Hall.

Cytel Software Corporation, 2000. EGRET for Windows, User manual. Cambridge, USA.

Davidian, M., Gallant, A.R., 1992. Smooth Nonparametric Maximum Likelihood Estimation for Population Pharmacokinetics, with Application to Quinidine. *Journal of Pharmacokinetics and Biopharmaceutics* 20: 529-556.

Davidian, M., Giltinan, D.M., 1995. Nonlinear Models for Repeated Measurement Data. London: Chapman & Hall.

Davidian, M., Giltinan, D.M., 2003. Nonlinear models for repeated measurement data: an overview and update. *Journal of Agricultural, Biological, and Environmental Statistics* 8(4): 387-419.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm (c/r: P22-37). *Journal of the Royal Statistical Society*, Series B 39: 1-22.

Evans, M., Swartz, T., 1995. Methods for Approximating Integrals in Statistics with Special Emphasis on Bayesian Integration Problems. *Statistical Science* 10(3): 254-272.

Gallant, A.R., Nychka, W.D., 1987. Semi-Nonparametric Maximum Likelihood Estimation. *Econometrica* 55: 363-390.

Ge, Z., Bickel, J.P., Rice A.J., 2004. An Approximate Likelihood Approach to Nonlinear Mixed Effects Models via Spline Approximation. *Computational Statistics & Data Analysis* 46(4): 747-776.

Gilks, W.R., Richardson, S., Spiegelhalter, D.J., 1996. Markov Chain Monte Carlo in Practice. London: Chapman and Hall.

Golub, G.H., 1973. Some Modified Matrix Eigenvalue Problems. *SIAM Review* 15: 318-334.

Golub, G.H., Welsch, J.H., 1969. Calculation of Gaussian Quadrature Rules. *Mathematics of Computation* 23: 221-230.

Harville, D.A., 1977. Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. *Journal of the American Statistical Association* 72: 320-340.

Hedeker, D., Gibbons, R.D., 1996a. MIXOR: A Computer Program for Mixed-Effects Ordinal Regression Analysis. *Biometrics* 50: 933-944.

Hedeker, D., Gibbons, R.D., 1996b. MIXREG: A Computer Program for Mixed-Effects Regression Analysis with Autocorrelated Errors. *Computer Methods and Programs in Biomedicine* 49: 229-252.

Hedeker, D., Marcantonio, R., Pechnyo, M., 2000. Chapter 2: Mixed Regression, in SYSTAT 10, Statistics II.

Insightful Corporation, 2007. S-PLUS 8 for Windows Guide to Statistics. Seattle, WA, Insightful Corporation.

Kauermann, G., Xu, R., Vaida, F., 2008. Stacked Laplace-EM algorithm for duration models with time-varying and random effects. *Computational Statistics & Data Analysis* 52, 2514-2528.

Kuhn, E., Lavielle, M., 2005. Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics & Data Analysis* 49, 1020-1038.

Laird, N.M., Ware, J.H., 1982. Random-Effects Models for Longitudinal Data. *Biometrics* 38: 963-974.

Landau, S., Everitt, B.S., 2004. A Handbook of Statistical Analyses using SPSS. Boca Raton: Chapman & Hall.

Lee, Y., Nelder, J.A., 1996. Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society*, Series B 58, 619-678.

Leonard, T., Hsu, J.S.J., Tsui, K.W., 1989. Bayesian Marginal Inference. *Journal of the American Statistical Association* 84: 1051-1058.

Lindstrom, M.J., Bates, D.M., 1990. Nonlinear Mixed Effects Models for Repeated Measures Data. *Biometrics* 46: 673-687.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., Schabenberger, O., 2006. SAS System for Mixed Models. Cary, NC: SAS Institute, Inc.

Longford, N.T., 1993. Random Coefficient Models. Oxford: Clarendon Press.

Mallet, A., 1986. A Maximum Likelihood Estimation Method for Random Coefficient Regression Models. *Biometrika* 73(3): 645-656.

Mallet, A., Mentre, F., Steimer, J.-L., Lokiec, F., 1988. Nonparametric Maximum Likelihood Estimation for Population Pharmacokinetics, with Applications to Cyclosporine. *Journal of Pharmacokinetics and Biopharmaceutics* 16: 311-327.

Mauritsen, R.H., 1984. Logistic Regression with Random Effects. Ph.D. thesis, Department of Biostatistics, University of Washington.

Naylor, J.C., Smith, A.F.M., 1982. Applications of a Method for the Efficient Computation of Posterior Distributions. *Applied Statistics* 31: 214-225

Noh, M., Lee, Y., 2008. Hierarchical-likelihood approach for nonlinear mixed-effects models. *Computational Statistics & Data Analysis* 52: 3517-3527.

Panhard, X., Samson, A., 2009. Extension of the SAEM algorithm for nonlinear mixed models with 2 levels of random effects. *Biostatistics* 10: 121-135.

Pinheiro, J.C., Bates, D.M., 1995. Approximations to the Log-likelihood Function in the Nonlinear Mixed-Effects Model. *Journal of Computational and Graphical Statistics* 4: 12-35.

Pinheiro, J.C., Bates, D.M., 2000. Mixed-Effects Models in S and S-PLUS. New York: Springer-Verlag.

R Development Core Team, 2009. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org

Rabe-Hesketh, S., Everitt, B., 2004. A Handbook of Statistical Analyses Using Stata, 3rd edition. Chapman & Hall/CRC.

Rasbash, J., Steele, F., Browne, W., 2003. A User's Guide to MLwiN Version 2.0. Institute of Education, London.

Raudenbush, S.W., Bryk, A., Cheong, Y.F., Congdon, R., 2001. HLM 5, Hierarchical Linear and Nonlinear Modelling. SSI, Lincolnwood.

SAS Institute, 2004. SAS OnlineDoc, Version 9.1.3, Cary, NC: SAS Institute Inc.

Schabenberger, O., Pierce, J.F., 2001. Contemporary Statistical Models for the Plant and Soil Sciences. CRC Press.

Sheiner, L.B., Beal, S.L., 1980. Evaluation of Methods for Estimating Population Pharmacokinetic Parameters. I. Michaelis-Menten Model: Routine Clinical Pharmacokinetic Data. *Journal of Pharmacokinetics and Biopharmaceutics* 8: 553-571.

Shun, Z., McCullagh, P., 1995. Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society*, Series B 57, 749-760.

Skaug, H., Fournier, D., 2006. Random effects in AD Model Builder (ADMB-RE) User Guide. Otter Research Ltd., Canada.

Spiegelhalter, D.J., Thomas, A., Best, N.G., Lunn, D., 2003. WinBUGS Version 1.4 Users Manual. MRC Biostatistics Unit, Cambridge. URL http://www.mrc-bsu.cam.ac.uk/bugs

The MIXED/NLMIXED Procedure. 1999. SAS/STAT User's Guide, SAS On-Line Document. SAS Institute Inc.

Thisted, R.A., 1988. Elements of Statistical Computing: Numerical computation. CRC Press.

Tierney, L., Kadane, J.B., 1986. Accurate Approximations for Posterior Moments and Densities. *Journal of the American Statistical Association* 81(393): 82-86.

Tierney, L., Kass, R.E., Kadane, J.B., 1989. Fully exponential Laplace Approximations to Expectations and Variances of Nonpositive Functions. *Journal of the American Statistical Association* 84: 710-716.

Vaida F., Fitzgerald A.P., DeGruttola V., 2007. Efficient hybrid EM for linear and nonlinear mixed effects models with censored response. *Computational Statistics & Data Analysis* 51, 5718-5730.

Verbeke, G., Molenberghs, G., 2000. Linear Mixed Models for Longitudinal Data. New York: Springer.

Vonesh, E.F., 1996. A Note on the Use of Laplace's Approximation for Nonlinear Mixed-Effects Models. *Biometrika* 83: 447-452.

Vonesh, E.F., Chinchilli, V., 1997. Linear and Nonlinear Models for the Analysis of Repeated Measurements. New York: Marcel Dekker Inc.

Vonesh, E.F., Wang, H., Nie, L., Majumdar, D., 2002. Conditional Second-Order Generalized Estimating Equations for Generalized Linear and Nonlinear Mixed-Effects Models. *Journal of the American Statistical Association* 97: 271-283.

VSN International Ltd., 2006. GenStat User Manual.

Wakefield, J.C., 1993. The Bayesian Analysis of Population Pharmacokinetic Models. Technical Report TR-93-11, Statistics Section, Imperial College.

Wakefield, J.C., Smith, A.F.M., Racine-Poon, A., Gelfand, A.E., 1994. Bayesian analysis of linear and nonlinear population models by using the Gibbs sampler. *Applied Statistics* 43, 201-221.

Walker, S.G., 1996. An EM Algorithm for Nonlinear Random Effects Models. *Biometrics* 52: 934-944.

Wang, J., 2007. EM algorithms for nonlinear mixed effects models. *Computational Statistics & Data Analysis* 51, 3244-3256.

Wolfinger, R., 1993. Laplace's approximation for nonlinear mixed models. *Biometrika* 80, 791-795.

Wolfinger, R.D., Lin, X., 1997. Two Taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis* 25, 465-490.

Yang, M., Goldstein, H., 2003. Modelling survival data in MLwiN 1.20. Institute of Education, London.

CHAPTER 3

REML ESTIMATION IN NONLINEAR MIXED EFFECTS MODELS VIA THE FULLY EXPONENTIAL LAPLACE APPROXIMATION EM ALGORITHM

3.0 Abstract

A new EM algorithm using the fully exponential Laplace method to approximate the conditional expectations of the complete data sufficient statistics in the E-step for obtaining restricted maximum likelihood estimates in nonlinear mixed effects models is developed. The main advantages of this approach are its combination of computational efficiency (preserving the numerically simplicity of Laplace approximation) and great estimation accuracy (giving an error of order $O(1/n^2)$ for estimating variance-covariance components). Four simulation studies have been conducted to evaluate the accuracy of the new approach and compare it with the Laplace approximation as well as four different linearization methods. Of all approximations considered in the paper, our proposed EM algorithm is the only one that gives unbiased or close-to-unbiased (%Bias < 1%, Ratkowsky 1983) estimates for all the fixed effects and variance-covariance components with 95% confidence interval coverages similar to the nominal values for all the fixed effects.

Keywords: Nonlinear mixed effects model; EM; Laplace; Restricted maximum likelihood

3.1 Introduction

Nonlinear mixed effects models have been used in numerous fields, such as pharmacokinetics, biology, medicine, and economics for analyzing repeated measure data. Estimation in nonlinear mixed effects models, typically, cannot be performed directly because the likelihood of these models has no closed-form expression due to the random effects entering nonlinearly in the models. Different approximation methods have been proposed to circumvent this problem. The most popular approximation methods used to estimate the parameters in nonlinear mixed effects models are linearization methods, which are based on using a first-order Taylor series expansion to approximate the nonlinear model function around zero that is the expected value of the random effects (Sheiner and Beal, 1980; Wolfinger and Lin, 1997, zero-expansion method), or around the current estimates of the random effects (Lindstrom and Bates, 1990; Wolfinger and Lin, 1997, EBLUP-expansion method), and then maximize the likelihood corresponding to the resulting approximate model. Linearization methods are computationally simple and have been implemented in a number of software packages such as NONMEM (Beal and Sheiner, 1992), the %nlinmix macro and the nlmixed procedure in SAS (SAS Institute, 2004), and the nlme function (Pinheiro and Bates, 2000) in both S-PLUS (Insightful Corporation, 2007) and R (R development core team, 2009).

Although linearization methods are popular and numerically simple, they can produce substantial bias in parameter estimation when the number of observations for each subject is small or the variability of the random effects is large (Ge, Bickel and Rice, 2004). This has motivated the researchers to use more accurate methods like Laplace approximation,

numerical integration methods such as Gaussian quadrature (Pinheiro and Bates, 1995), and Monte Carlo methods such as importance sampling (Pinheiro and Bates, 1995) and Markov Chain Monte Carlo technique (Wakefield, Smith, Racine-Poon, and Gelfand, 1994) to improve the estimation in nonlinear mixed effects models. Laplace approximation methods are based on using a second-order Taylor series expansion to integrate out either the random effects (Pinheiro and Bates, 1995; Vonesh, 1996), or both the random effects and the fixed effects by assuming a flat prior for the fixed effects (Wolfinger, 1993) from the marginal likelihood of nonlinear mixed models. While Laplace approximation methods provide more accurate estimates than linearization methods and are computationally efficient, the maximum likelihood (ML) Laplace approximation procedure, the one that integrates out the random effects only, can introduce a bias for mixed effects models when the number of subjects is small (Breslow and Lin, 1995; Shun and McCullagh, 1995; Kauermann, Xu, and Vaida, 2008). The performance of the restricted maximum likelihood (REML) Laplace approximation, the one that integrates out both the random and fixed effects, still needs further investigation. Numerical integration methods and Monte Carlo methods are often referred as "exact" methods in statistical literature in the sense that they can be made as accurate as desired by taking sufficient large number of grid points (numerical integration methods) or simulated samples (Monte Carlo methods). The "exact" approach works extremely well only for single-level nonlinear mixed models with small number of random effects (e.g., one or two), but can become computationally great intensive as the number of random effects increases (Vonesh, Wang, Nie, and Majumdar, 2002).

As an alternative to the direct approximate to the marginal likelihood of nonlinear mixed effects models, various simulated EM algorithms for nonlinear mixed models have received increased interest in the statistical literature. The Monte Carlo version EM algorithms approximate conditional expectations in the E-step by Monte Carlo integration and simulated samples are either from the distribution of the random effects (Walker, 1996) or via importance sampling from a mixture distribution that is simple in the form, easy to sample from, and efficient (Wang, 2007). The stochastic approximation version EM algorithm (Kuhn and Lavielle, 2005) replaces the E-step by a simulation step and a stochastic approximation step and then simulation samples are obtained from a Markov Chain Monte Carlo procedure. Like the Monte Carlo methods for direct approximation to the marginal likelihood of nonlinear mixed effects models, the computation of the simulated EM algorithms can also be challenging as the number of random effects increases.

The problems of the current estimation methods for nonlinear mixed effects models (either producing biased estimates or numerically intensive) form the motivation of this study. In this paper, we developed a new, efficient, and accurate EM algorithm, fully exponential Laplace approximation EM algorithm (FELA-EM), for obtaining restricted maximum likelihood (REML) estimates in parametric nonlinear mixed effects models. In the E-step of the FELA-EM algorithm, the fully exponential Laplace method (Tierney and Kadane 1986; Tierney, Kass, and Kadane 1989) is used to approximate the conditional expectations of the complete data sufficient statistics. The resulting approximations are generally as accurate as those based on third-order expansions and

requiring the evaluation of third derivatives. We prefer REML estimation for variance parameters because it reduces bias by taking account of the degrees of freedom lost in estimating the fixed effects. The bias of the maximum likelihood (ML) estimators can be substantial when the number of subjects is small in the nonlinear mixed effects model. Similar to Wolfinger (1993) and Noh and Lee (2008), we obtain REML estimates for variance parameters by integrating out the fixed effects from the conditional likelihood of nonlinear mixed models assuming a flat prior for the fixed effects. We show that the FELA-EM algorithm gives more accurate estimates, an error of order $O(1/n^2)$, than those from Laplace approximation, an error of order $O(1/n)$, while preserving the numerically simplicity of Laplace approximation. Thus, the FELA-EM algorithm is computationally much simpler than numerical integration and Monte Carlo methods.

This paper is organized as follows. In Section 3.2, we present the nonlinear mixed model and the likelihood of the model. The FELA-EM algorithm is introduced in Section 3.3. In Section 3.4, we compare the proposed EM algorithm with the ML version of Laplace approximation as well as four linearization methods based on simulation data. We give our overall conclusions in Section 3.5 and summarize the paper in Section 3.6.

3.2 Model and likelihood

We consider the following nonlinear mixed effects model:

$$y_{ij} = f(\boldsymbol{\beta}, \boldsymbol{b}_i) + \varepsilon_{ij}, \quad i = 1,...,M, \quad j = 1,...,n_i, \tag{3.2.1}$$

where y_{ij} is the jth observation on the ith subject, f is a nonlinear function, $\boldsymbol{\beta}$ is a p-dimensional vector of fixed effects, $\boldsymbol{b}_i$ is a q-dimensional random effects vector associated with the ith subject (not varying with j) and assumed i.i.d. normal with mean $\mathbf{0}$ and variance-covariance matrix $\boldsymbol{D}$, ε_{ij} is the error and assumed i.i.d. $N(0,\sigma^2)$, M is the number of subjects, and n_i is the number observations on the ith subject. It is further assumed that b_i and ε_{ij} are independent.

Because the random effects $\boldsymbol{b}$ are unobserved quantities, ML estimation in (3.2.1) is based on the marginal density of $\boldsymbol{y}$, which is calculated as

$$p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{D}, \sigma^2) = \int p(\boldsymbol{y}, \boldsymbol{b} \mid \boldsymbol{\beta}, \boldsymbol{D}, \sigma^2) d\boldsymbol{b} = \int p(\boldsymbol{y} \mid \boldsymbol{b}, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{b} \mid \boldsymbol{D}) d\boldsymbol{b}. \qquad (3.2.2)$$

Similarly, by assuming a flat prior for the fixed effects $\boldsymbol{\beta}$, REML estimation for variance-covariance components $\boldsymbol{D}$ and σ^2 in (3.2.1) can be obtained by integrating out both the fixed effects and the random effects from the joint density of $\boldsymbol{y}$, $\boldsymbol{\beta}$, and $\boldsymbol{b}$, which is given by

$$p(\boldsymbol{y} \mid \boldsymbol{D}, \sigma^2) = \int p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b} \mid \boldsymbol{D}, \sigma^2) d\boldsymbol{\beta} d\boldsymbol{b} = \int p(\boldsymbol{y} \mid \boldsymbol{b}, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{b} \mid \boldsymbol{D}) d\boldsymbol{\beta} d\boldsymbol{b}. \qquad (3.2.3)$$

Since the model function f can be nonlinear in the fixed and random effects, both integrals (3.2.2) and (3.2.3) generally do not have a closed-form expression.

3.3 FELA-EM algorithm for REML estimates of variance-covariance components

The EM algorithm introduced by Dempster, Laird, and Rubin (1977) has become a popular approach for finding maximum likelihood estimates in incomplete data problems due to its simplicity and stability. It is an iterative method that alternates between an expectation step (E-step) and a maximization step (M-step). Let $\boldsymbol{y}_{obs}$ and $\boldsymbol{y}_{mis}$ represent the observed and missing data, respectively. Let $c(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis})$ represent the complete data and $\boldsymbol{\theta}$ be the vector of parameters to be estimated. On the $(k+1)$th iteration, the E-step generally computes the expectation of the complete data log-likelihood $\ell(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis}; \boldsymbol{\theta})$ conditional on the observed data $\boldsymbol{y}_{obs}$ at the current estimates of the parameters $\boldsymbol{\theta}^{(k)}$ (or the initial values for the first iteration),

$$Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)}) = E[\ell(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis}; \boldsymbol{\theta})\big|\boldsymbol{y}_{obs}; \boldsymbol{\theta}^{(k)}].$$

The M-step is then to find $\boldsymbol{\theta}^{(k+1)}$ to maximize $Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)})$,

$$Q(\boldsymbol{\theta}^{(k+1)}\big|\boldsymbol{\theta}^{(k)}) \geq Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)}),$$

for all $\boldsymbol{\theta}$ in the parameter space $\boldsymbol{\Omega}$. The two steps are repeated until convergence. When the complete data density belongs to the exponential family, the E-step is simplified to compute the expected sufficient statistics of the complete data. The EM algorithm moves

to a better point at each iteration and the log-likelihood of the observed data does not decrease after an EM iteration.

For the nonlinear mixed effects model (3.2.1), assume a flat prior for the fixed effects $\boldsymbol{\beta}$ and consider $c(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b})$ and $c(\boldsymbol{\beta},\boldsymbol{b})$ as the complete data and the missing data, respectively. Let $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ represent the parameters for which REML estimates are required. The E-step and M-step of the FELA-EM algorithm are described in sections 3.3.1 and 3.3.2 respectively.

3.3.1 E-step

The E-step gives the conditional expectation of the complete data log-likelihood $\ell(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta})$,

$$Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)}) = \int \ell(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}) p(\boldsymbol{\beta},\boldsymbol{b}\big|\boldsymbol{y};\boldsymbol{\theta}^{(k)}) d\boldsymbol{\beta} d\boldsymbol{b}, \tag{3.3.1}$$

where

$$\ell(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{M}\left\|\boldsymbol{y}_i - f(\boldsymbol{\beta},\boldsymbol{b}_i)\right\|^2 - \frac{M}{2}\log(2\pi|\boldsymbol{D}|) - \frac{1}{2}\sum_{i=1}^{M}\boldsymbol{b}_i^T\boldsymbol{D}^{-1}\boldsymbol{b}_i \tag{3.3.2}$$

and $N = \sum_{i=1}^{M} n_i$ is the total number of observations. The density of the missing data $c(\boldsymbol{\beta}, \boldsymbol{b})$ conditional on the observed data $\boldsymbol{y}$ at $\boldsymbol{\theta}^{(k)}$, $p(\boldsymbol{\beta}, \boldsymbol{b} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(k)})$, can be further written as

$$p(\boldsymbol{\beta}, \boldsymbol{b} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(k)}) = \frac{p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b} \big| \boldsymbol{\theta}^{(k)})}{p(\boldsymbol{y} \big| \boldsymbol{\theta}^{(k)})} = \frac{\exp\{\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})\}}{\int \exp\{\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})\} d\boldsymbol{\beta} d\boldsymbol{b}}. \tag{3.3.3}$$

It can be seen from (3.3.2) that the density of the complete data $c(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b})$ is from the exponential family and the sufficient statistics of the complete data for σ^2 and $\boldsymbol{D}$ are $R = \sum_{i=1}^{M} \left\| \boldsymbol{y}_i - f(\boldsymbol{\beta}, \boldsymbol{b}_i) \right\|^2 = \left\| \boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b}) \right\|^2$ and $S_{m,n} = \sum_{i=1}^{M} b_{i,m} b_{i,n} = \boldsymbol{b}_m' \boldsymbol{b}_n$ for $m, n = 1, ..., q$, respectively. Thus, the E-step is simplified to compute the expectations of R and $S_{m,n}$ conditional on $\boldsymbol{y}$ at $\boldsymbol{\theta}^{(k)}$ that are given by

$$E(R \big| \boldsymbol{y}; \boldsymbol{\theta}^{(k)}) = \frac{\int \left\| \boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b}) \right\|^2 \exp\{\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})\} d\boldsymbol{\beta} d\boldsymbol{b}}{\int \exp\{\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})\} d\boldsymbol{\beta} d\boldsymbol{b}}, \tag{3.3.4}$$

and

$$E(S_{m,n} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(k)}) = \frac{\int \boldsymbol{b}_m' \boldsymbol{b}_n \exp\{\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})\} d\boldsymbol{\beta} d\boldsymbol{b}}{\int \exp\{\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})\} d\boldsymbol{\beta} d\boldsymbol{b}}, \qquad m, n = 1, ..., q, \tag{3.3.5}$$

where

$$\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)}) = -\frac{1}{2\{\sigma^{(k)}\}^2}\sum_{i=1}^{M}\left\|\boldsymbol{y}_i - f(\boldsymbol{\beta},\boldsymbol{b}_i)\right\|^2 - \frac{1}{2}\sum_{i=1}^{M}\boldsymbol{b}_i^T\{\boldsymbol{D}^{(k)}\}^{-1}\boldsymbol{b}_i,$$

and the common terms that appear in both the numerator and denominator in (3.3.4) and (3.3.5) and do not contain either $\boldsymbol{\beta}$ or $\boldsymbol{b}$ are cancelled. The integrals in both the numerator and the denominator of (3.3.4) and (3.3.5) generally cannot be computed analytically because both R and $\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})$ can be nonlinear in the fixed and random effects. In this paper, we approximate both (3.3.4) and (3.3.5) by the fully exponential Laplace method that will be presented in Section 3.3.3.

3.3.2 M-step

The M-step is to find $\theta^{(k+1)}$ by solving the equation

$$\frac{\partial}{\partial\boldsymbol{\theta}}Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(k)}) = \frac{\partial}{\partial\boldsymbol{\theta}}\int \ell(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta})p(\boldsymbol{\beta},\boldsymbol{b}\big|\boldsymbol{y};\boldsymbol{\theta}^{(k)})d\boldsymbol{\beta}d\boldsymbol{b} = 0. \tag{3.3.5}$$

By allowing differentiation under the integral sign, the unique solution to (3.3.5) is given by

$$\{\sigma^{(k+1)}\}^2 = \frac{1}{N}E(R\big|\boldsymbol{y};\boldsymbol{\theta}^{(k)}), \tag{3.3.6}$$

and

$$D_{m,n}^{(k+1)} = \frac{1}{M} E(S_{m,n} \Big| \boldsymbol{y}; \boldsymbol{\theta}^{(k)}), \qquad m, n = 1,..., q. \tag{3.3.7}$$

See Walker (1996) and Wang (2007) for a more detailed derivation. Therefore, once the conditional expectations of the sufficient statistics $E(R\Big|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ and $E(S_{m,n}\Big|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ for $m, n = 1,..., q$ are obtained, the process of the M-step is straightforward.

3.3.3 Fully exponential Laplace approximation

The fully exponential Laplace approximation to the ratio of two related integrals introduced by Tierney and Kadane (1986) is given by

$$\begin{aligned} \frac{\int g(\boldsymbol{\varphi}) \exp\{n\ell(\boldsymbol{\varphi})\} d\boldsymbol{\varphi}}{\int \exp\{n\ell(\boldsymbol{\varphi})\} d\boldsymbol{\varphi}} &= \frac{\int \exp\{n\ell^*(\boldsymbol{\varphi})\} d\boldsymbol{\varphi}}{\int \exp\{n\ell(\boldsymbol{\varphi})\} d\boldsymbol{\varphi}} \\ &\approx \left(\frac{\det\{-\partial^2 \ell(\hat{\boldsymbol{\varphi}}) / \partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'\}}{\det\{-\partial^2 \ell^*(\hat{\boldsymbol{\varphi}}^*) / \partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'\}} \right)^{1/2} \exp\{n\ell^*(\hat{\boldsymbol{\varphi}}^*) - n\ell(\hat{\boldsymbol{\varphi}})\}, \end{aligned} \tag{3.3.8}$$

where $g(\boldsymbol{\varphi})$ is a positive scalar function, $\ell^*(\boldsymbol{\varphi}) = \log\{g(\boldsymbol{\varphi})\}/n + \ell(\boldsymbol{\varphi})$, and $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\varphi}}^*$ maximize ℓ and ℓ^*, respectively. Although the errors in the Laplace approximations to the two integrals (numerator and denominator) in (3.3.8) are of order $O(1/n)$, the error in the ratio (3.3.8) is of order $O(1/n^2)$ due to the cancellation of the similar error terms in the approximation to the two integrals. The limitation of the Laplace approximation in (3.3.8) is that it only applies to positive functions. Thus, it is not suitable for our problem

because the covariance parameters can be negative and even the variance components are not necessarily strictly positive and they can take values close to zero or zero. Tierney, Kass and Kadane (1989) generalize the Laplace approximation in (3.3.8) so that $g(\boldsymbol{\varphi})$ can take on negative values. Their approach is to first approximate the moment generating function of $g(\boldsymbol{\varphi})$ so that it is strictly positive, and then approximate the ratio in (3.3.8) by evaluating the first derivative of the approximation to the moment generating function at 0. They showed that the accuracy of the moment generating function approach is also of order $O(1/n^2)$.

We now show how to apply the moment generating function approach to estimate the conditional expectations $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ and $E(S_{m,n}|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ for $m,n=1,...,q$ given in (3.3.4) and (3.3.5), respectively.

The moment generating function of $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ is given by

$$M_R(t)=\frac{\int \exp\{t\|\boldsymbol{y}-f(\boldsymbol{\beta},\boldsymbol{b})\|^2+\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})\}d\boldsymbol{\beta}d\boldsymbol{b}}{\int \exp\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})\}d\boldsymbol{\beta}d\boldsymbol{b}}, \tag{3.3.9}$$

with $\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})$ as in (3.3.5). Applying the fully exponential Laplace approximation (3.3.8) to (3.3.9) yields

$$\widetilde{M}_R(t) = \left(\frac{\det[-\partial^2 \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}; \boldsymbol{\theta}^{(k)}) / \partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}']}{\det[-\partial^2 \{t \left\| \boldsymbol{y} - f(\hat{\boldsymbol{\beta}}^*, \hat{\boldsymbol{b}}^*) \right\|^2 + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}^*, \hat{\boldsymbol{b}}^*; \boldsymbol{\theta}^{(k)})\} / \partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}']} \right)^{1/2} \tag{3.3.10}$$

$$\times \exp\{t \left\| \boldsymbol{y} - f(\hat{\boldsymbol{\beta}}^*, \hat{\boldsymbol{b}}^*) \right\|^2 + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}^*, \hat{\boldsymbol{b}}^*; \boldsymbol{\theta}^{(k)}) - \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}; \boldsymbol{\theta}^{(k)})\},$$

where $\hat{\boldsymbol{\varphi}}^* = c(\hat{\boldsymbol{\beta}}^*, \hat{\boldsymbol{b}}^*)$ maximizes $t\|\boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b})\|^2 + \hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ and $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}})$ maximizes $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$. Then $E(R|\boldsymbol{y}; \boldsymbol{\theta}^{(k)})$ can be approximated by $\partial \widetilde{M}_R(t) / \partial t$ evaluated at 0. Since the analytic differentiation of $\widetilde{M}_R(t)$ with respect to t requires the third derivatives of $\|\boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b})\|^2$ and $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ with respect to $\boldsymbol{\varphi}$ and that is substantial work, we apply the numerical differentiation approach to approximate $E(R|\boldsymbol{y}; \boldsymbol{\theta}^{(k)})$,

$$E(R|\boldsymbol{y}; \boldsymbol{\theta}^{(k)}) = \frac{\partial \widetilde{M}_R(0)}{\partial t} \approx \frac{\widetilde{M}_R(\delta) - \widetilde{M}_R(-\delta)}{2\delta},$$

for some small δ. We choose $\delta = 10^{-8}$.

Similarly, one can show that the fully exponential Laplace approximation to the moment generating function of $E(S_{m,n}|\boldsymbol{y}; \boldsymbol{\theta}^{(k)})$ for $m, n = 1, ..., q$ is given by

$$\widetilde{M}_{S_{m,n}}(t) = \left(\frac{\det[-\partial^2 \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}; \boldsymbol{\theta}^{(k)}) / \partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}']}{\det[-\partial^2 \{t\boldsymbol{b}_m' \boldsymbol{b}_n + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}^{**}, \hat{\boldsymbol{b}}^{**}; \boldsymbol{\theta}^{(k)})\} / \partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}']} \right)^{1/2}$$
$$\times \exp\{t\boldsymbol{b}_m' \boldsymbol{b}_n + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}^{**}, \hat{\boldsymbol{b}}^{**}; \boldsymbol{\theta}^{(k)}) - \hbar(\boldsymbol{y}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}; \boldsymbol{\theta}^{(k)})\}, \tag{3.3.11}$$

where $\hat{\boldsymbol{\varphi}}^{**} = c(\hat{\boldsymbol{\beta}}^{**}, \hat{\boldsymbol{b}}^{**})$ maximizes $t\boldsymbol{b}_m' \boldsymbol{b}_n + \eta(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ and $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}})$ maximizes $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$.

As pointed out by Tierney and Kadane (1986), once $\hat{\boldsymbol{\varphi}}$, the maximum of $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$, has been determined, it can be used as starting values to find $\hat{\boldsymbol{\varphi}}^{*}$ and $\hat{\boldsymbol{\varphi}}^{**}$, the maximum of $t\|\boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b})\|^2 + \hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ and $t\boldsymbol{b}_m' \boldsymbol{b}_n + \hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ in (3.3.10) and (3.3.11), respectively. Generally, the number of iterations needed to find $\hat{\boldsymbol{\varphi}}^{*}$ and $\hat{\boldsymbol{\varphi}}^{**}$ from $\hat{\boldsymbol{\varphi}}$ is quite small. Replacing $\hat{\boldsymbol{\varphi}}^{*}$ and $\hat{\boldsymbol{\varphi}}^{**}$ by two Newton steps from $\hat{\boldsymbol{\varphi}}$ is usually sufficient. Thus, the computational requirements of the fully exponential Laplace approximation are comparatively minimal.

In summary, the FELA-EM algorithm takes the following steps:

1) Initialize $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}) = c(\sigma^2, \boldsymbol{D}) = \boldsymbol{\theta}_0$.

2) E-step:

   a) Find $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}})$ to maximize $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$

b) Maximize $t\|\boldsymbol{y}-f(\boldsymbol{\beta},\boldsymbol{b})\|^2+\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})$ and $t\boldsymbol{b}_m'\boldsymbol{b}_n+\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})$ by replacing $\hat{\boldsymbol{\varphi}}^{*}=c(\hat{\boldsymbol{\beta}}^{*},\hat{\boldsymbol{b}}^{*})$ and $\hat{\boldsymbol{\varphi}}^{**}=c(\hat{\boldsymbol{\beta}}^{**},\hat{\boldsymbol{b}}^{**})$ with two Newton steps from $\hat{\boldsymbol{\varphi}}$, respectively.

c) Approximate the conditional expectations $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ and $E(S_{m,n}|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ for $m,n=1,...,q$ with the numerical differentiation approach.

3) M-step: Obtain $\boldsymbol{\theta}^{(k+1)}=c(\{\sigma^{(k+1)}\}^2,\boldsymbol{D}^{(k+1)})$ according to (3.3.6) and (3.3.7).

4) Repeat steps 2 and 3 until convergence.

We stop the algorithm when either the difference between two successive log-likelihood of model (3.2.1) or the absolute/relative changes in $\boldsymbol{\theta}$ are less than 10^{-6}. The convergence rate of the standard EM algorithm is slow (McLachlan and Krishnan, 2008). When the absolute/relative changes in $\boldsymbol{\theta}$ are less than 10^{-3}, we switch to the accelerated version of the EM algorithm, algorithm QN1, proposed by Jamshidian and Jennrich (1997). The QN1 algorithm is based on a quasi-Newton method, Broyden's method, for solving nonlinear equations and minimizing functions. Algorithm QN1 is easy to implement and its speed of convergence can be 19-87 times faster than that of the standard EM algorithm. The idea of Algorithm QN1 is as follows. For solving the system of equations G(x) = 0, Broyden's method uses search directions of the form $-A_iG(x_i)$. In Newton's method A_i is the inverse of the Jacobian of G(x), while in Broyden's method an approximation to the Jacobian of G(x) is used. The updates to A_i in Broyden's method are chosen to satisfy a secant condition and the inverse Jacobian updating formula is given by

$$A_{i+1} = A_i + (s'A_ih)^{-1}(s - A_ih)(s'A_i) \quad (3.3.12)$$

where $s = x_{i+1} - x_i$ and $h = G(x_{i+1}) - G(x_i)$. Let $M(\theta_0)$ be the value of θ given by one standard EM update from the current value θ_0; that is, the value maximizing $Q(\theta|\theta_0)$. The objective of the EM algorithm can be expressed as finding a solution to $g(\theta) = M(\theta) - \theta = 0$, so here $g(\theta)$ takes the role of $G(x)$ in the above discussion. In Jamshidian and Jennrich's approach, the Broyden update step is applied to the $g(\theta)$ function, as follows. First initialize by setting $\theta = \theta_0$, $g_0 = g(\theta_0)$, and $A = -I$ (the negative of an identity matrix). Then

1. Compute $s = -Ag_0$ and $h = g(\theta + s) - g_0$;
2. Update A using (3.3.12) and s and h from step 1;
3. Replace θ by $\theta + s$ and g_0 by $g_0 + h$, and return to step 1, repeating until convergence.

3.3.4 Calculating the information matrix

The EM algorithm does not automatically provide the information matrix from which the standard errors of the estimates can be obtained. Let $\hat{\boldsymbol{\theta}}$ be the estimates of $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ at convergence. The observed information matrix of $\hat{\boldsymbol{\theta}}$ can be obtained by: 1) directly approximating the REML version observed data log-likelihood $\log\{p(\boldsymbol{y} \mid \boldsymbol{D}, \sigma^2)\}$ by the standard Laplace method, where $p(\boldsymbol{y} \mid \boldsymbol{D}, \sigma^2)$ is the observed data density given in (3.2.3), and 2) maximizing the approximated log-likelihood by one quasi-Newton step from $\hat{\boldsymbol{\theta}}$.

3.3.5 Estimating the fixed and random effects

The FELA-EM algorithm proposed above is a REML method for estimating variance and covariance parameters $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ in the nonlinear mixed model (3.2.1) and it does not provide estimates for both the fixed effects $\boldsymbol{\beta}$ and the random effects $\boldsymbol{b}$. To estimate $\boldsymbol{\beta}$ and $\boldsymbol{b}$ in (3.2.1), we first apply the standard Laplace approximation to the ML version observed data log-likelihood $\log\{p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{D}, \sigma^2)\}$, where $p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{D}, \sigma^2)$ is the observed data density given in (3.2.2). We then maximize the approximated log-likelihood with respect to both $\boldsymbol{\beta}$ and $\boldsymbol{b}$ by holding $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ at $\hat{\boldsymbol{\theta}}$, the estimates at convergence from the FELA-EM algorithm, to obtain the "REML estimates" of $\boldsymbol{\beta}$ and $\boldsymbol{b}$.

3.4 Comparing the approximations

In this section, we present a comparison of our proposed EM algorithm (FELA-EM-REML), the ML version Laplace approximation (Laplace-ML), and four linearization methods described in Section 3.1 via simulation studies. We restrict ourselves to two ML linearization methods, Sheiner and Beal (1980) and Lindstrom and Bates (1990), denoted as S-B-ML and L-B-ML, respectively, and two REML linearization methods proposed by Wolfinger and Lin (1997), denoted as ZERO-REML and EBLUP-REML. Two models are used in the simulation studies: a logistic model and a first-order compartment model. Both models are widely used in statistical literature to illustrate the fitting of nonlinear mixed models (Lindstrom and Bates, 1990; Pinheiro and Bates, 1995; Wolfinger and Lin, 1997; Kuhn and Lavielle, 2005; Wang, 2007). Each model is simulated under two conditions: small and large variances-covariances. For each setting of the model, 4,000 simulated data sets from four random number seeds (1,000 data sets/seed) are generated to avoid simulation bias and ML/REML estimates using the different approximation methods are obtained. The FELA-EM algorithm presented here is performed using the code written in R by the authors. The Laplace-ML approximation and L-B-ML method are implemented using the nlmer and nlme functions in R, respectively. The S-B-ML method is done using the nlmixed procedure in SAS while the two REML linearization methods (ZERO-REML and EBLUP-REML) use the %nlinmix macro in SAS.

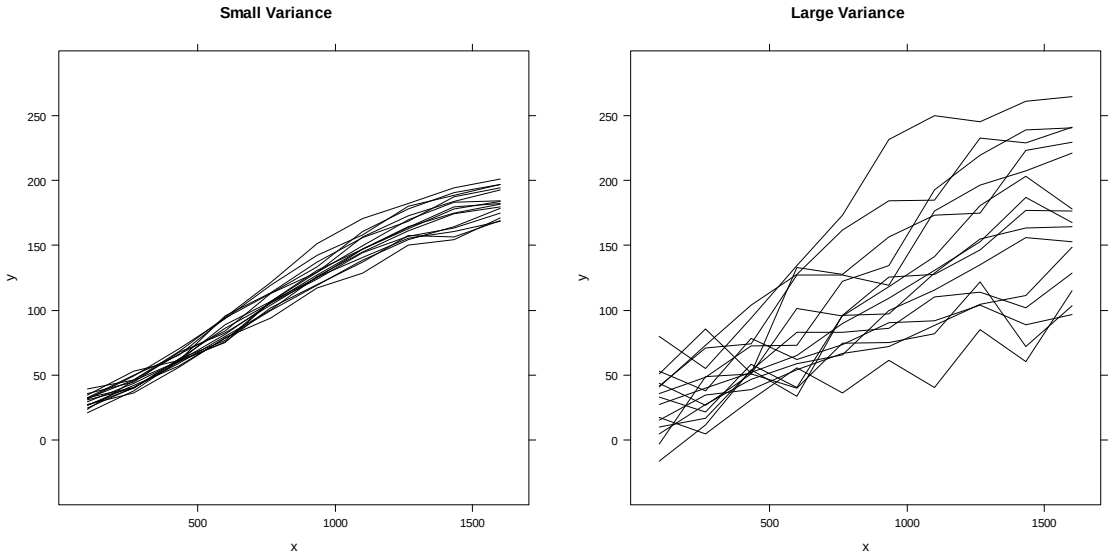
3.4.1 Logistic model

A 3-parameter logistic model with two random effects used by Pinheiro and Bates (1995) was used to generate the data. The values of fixed-effects parameters and covariate were also similar to those used by Pinheiro and Bates (1995). The nonlinear mixed model is given by

$$y_{ij} = \frac{\beta_1 + b_{i1}}{1 + \exp\{-[x_{ij} - (\beta_2 + b_{i2})]/\beta_3\}} + \varepsilon_{ij}, \qquad i = 1,...,M, \quad j = 1,...,n_i, \tag{3.4.1}$$

where $\boldsymbol{b}_i = (b_{i1}, b_{i2})^T$ are i.i.d. $N(0, \boldsymbol{D})$. The ε_{ij} are i.i.d. $N(0, \sigma^2)$ and independent of $\boldsymbol{b}_i$. We use $M = 15$, $n_i = 10$ for $i = 1,...,M$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^T = (200, 700, 350)^T$, and $\boldsymbol{x}_i = (100, 267, 433, 600, 767, 933, 1100, 1267, 1433, 1600)^T$. Two situations based on the covariance matrix are considered, denoted as small variance and large variance, respectively. For small variance, the settings are similar to those used by Pinheiro and Bates (1995), $\boldsymbol{D} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} 100 & -50 \\ -50 & 625 \end{bmatrix}$ and $\sigma^2 = 10$. For large variance, the values for $\boldsymbol{D}$ are similar to those used by Wolfinger and Lin (1997); we multiply each element of $\boldsymbol{D}$ and σ^2 for the small variance settings by 25, that is, $\boldsymbol{D} = \begin{bmatrix} 2500 & -1250 \\ -1250 & 15625 \end{bmatrix}$ and $\sigma^2 = 250$. Figure 3.1 shows sample profiles for each of the two settings of variance and covariance parameters.

Figure 3.1: Example of simulated logistic curves for small and large variance and covariance parameters



Tables 3.1 and 3.2 list the simulation results for the fixed-effects estimates in the logistic model for small and large variance-covariance components D and σ^2, respectively. Assuming $\hat{\theta}_r$ stands for a parameter estimate at the r^{th} simulation and θ_T for the true value of the parameter, the summary statistics for the parameters are defined as follows: Mean denotes the average of the estimates $\hat{\theta}_r$ across the 4000 simulations, i.e., $\sum_{r=1}^{4000} \hat{\theta}_r / 4000$; %Bias denotes the relative bias in estimating the parameter, i.e., $100 \times (\text{Mean} - \theta_T) / |\theta_T|$; RMSE denotes the square root of mean square error of the estimator, i.e., $\sqrt{\sum_{r=1}^{4000} (\hat{\theta}_r - \theta_T)^2 / 4000}$, which is a measure of accuracy that takes into account both bias and variability; 95% CI denotes a 95% confidence interval, i.e., $\text{Mean} \pm \text{Z}_\alpha \sqrt{s_{\hat{\theta}}^2 / 4000}$, where Z_α is the normal critical value corresponding to 95%

confidence level and $s_{\hat{\theta}}^2 = \sum_{r=1}^{4000} (\hat{\theta}_r - \theta_T)^2 / (4000-1)$. When a 95% CI does not cover the true value of the parameter, we conclude that the estimate is significantly biased at 5% level; and finally, %CVR denotes the observed coverage of the *t*-distribution based 95% confidence intervals computed using the model-based standard errors. Only those values for the fixed effects were presented in the paper because the standard errors for the variance-covariance parameters were not provided by either the nlmer function in R or the %nlinmix macro in SAS. The 95% coverage values marked with an asterisk are outside the interval (93.97, 96.03). The half-width of this interval is three times the binomial standard error, which is $[(95)(5)/4000]^{1/2}$ = 0.3446. When the variance-covariance components D and σ^2 are small, all six approximations considered provide good results for the fixed-effects parameters. Although the 95% confidence intervals show that the four linearization methods produce significantly biased estimates for some fixed effects, the %Bias are relatively small with a maximum absolute value 0.20% for L-B-ML approximation for estimating β_3. The two approximations which give unbiased estimates for all three fixed effects are FELA-EM-REML and Laplace-ML. For the observed 95% confidence interval coverages, all six approximations attain their nominal coverages for both β_1 and β_2, which are fixed effects associated with the random effects. The observed coverages are slightly different for β_3, which is the only one not associated with the random effects and enters the model nonlinearly. While both L-B-ML and FELA-EM-REML methods give values similar to their nominal ones, the coverages tend to increase from the nominal value for the other four methods (S-B-ML, Laplace-ML, ZERO-REML, and EBLUP-REML). When D and σ^2 increase, the results for the fixed-effects parameters are different. For β_1, which enters the nonlinear mixed model linearly

and is associated with the random effects, the two eblup-expansion linearization methods (L-B-ML and EBLUP-REML) underestimate the estimator by 1.90% and 1.83%, respectively while the other four approximations provide more accurate estimates – either unbiased (Laplace-ML) or close-to-unbiased (FELA-EM-REML, S-B-ML, and ZERO-REML) with a maximum absolute value of %Bias 0.32%. The results for β_2, which enters the model nonlinearly and is associated with the random effects, are similar to those for β_1. The two eblup-expansion linearization methods (L-B-ML and EBLUP-REML) again underestimate the estimator by 2.48% and 2.42%, respectively while the other four approximations give more accurate estimates – all close-to-unbiased with a maximum absolute value of %Bias 0.56% for Laplace-ML approximation. For β_3, which enters the model nonlinearly and is the only one not associated with the random effects, the estimates obtained from FELA-EM-REML and Laplace-ML approximations are more accurate than those from the four linearization methods. The two zero-expansion linearization methods (S-B-ML and ZERO-REML) significantly overestimate β_3 by 3% while the two eblup-expansion linearization methods (L-B-ML and EBLUP-REML) underestimate the estimator by more than 4%. Although FELA-EM-REML and Laplace-ML approximations give significantly positive biased estimates for β_3, the %Bias is 0.56% and 0.90%, respectively. That is more than three times smaller than those from the four linearization methods. For the 95% confidence interval coverages, the two eblup-expansion linearization methods (L-B-ML and EBLUP-REML) give considerable lower coverages for all three fixed effects. While Laplace-ML and the two zero-expansion linearization methods (S-B-ML and ZERO-REML) attain their nominal coverages for β_1 and β_2, their coverages for β_3 are significantly higher than the nominal

ones. The only approximation method which gives coverages similar to the nominal value for all three fixed effects is FELA-EM-REML. For each of the three fixed effects, the square root of mean square errors (RMSE) are relatively similar for all six approximations considered no matter the size of the variance-covariance components D and σ^2.

Table 3.1: Simulation results for the fixed effects in the logistic model for small variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI	%CVR
$\beta_1 = 200$					
S-B-ML	200.00	0.00	2.89	[199.91, 200.09]	94.49
L-B-ML	199.80†	-0.10	2.90	[199.71, 199.89]	94.36
Laplace-ML	199.99	-0.01	2.89	[199.90, 200.08]	94.37
ZERO-REML	200.00	0.00	2.89	[199.91, 200.09]	95.19
EBLUP-REML	199.81†	-0.10	2.90	[199.72, 199.90]	95.23
FELA-EM-REML	200.00	0.00	2.89	[199.91, 200.09]	94.50
$\beta_2 = 700$					
S-B-ML	699.78	-0.03	8.97	[699.50, 700.05]	95.52
L-B-ML	699.14†	-0.12	8.98	[698.87, 699.42]	95.39
Laplace-ML	700.06	0.01	8.97	[699.78, 700.34]	95.40
ZERO-REML	699.77	-0.03	8.98	[699.50, 700.05]	95.93
EBLUP-REML	699.16†	-0.12	8.97	[698.89, 699.44]	95.85
FELA-EM-REML	700.01	0.00	8.97	[699.73, 700.28]	95.43
$\beta_3 = 350$					
S-B-ML	350.42†	0.12	4.70	[350.27, 350.56]	96.52*
L-B-ML	349.31†	-0.20	4.70	[349.17, 349.46]	95.92
Laplace-ML	350.12	0.04	4.68	[349.98, 350.27]	96.62*
ZERO-REML	350.41†	0.12	4.71	[350.27, 350.56]	96.64*
EBLUP-REML	349.33†	-0.19	4.70	[349.18, 349.47]	96.05*
FELA-EM-REML	350.08	0.02	4.68	[349.93, 350.22]	95.98

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

* The %CVR is outside the interval (93.97, 96.03).

Table 3.2: Simulation results for the fixed effects in the logistic model for large variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI	%CVR
$\beta_1 = 200$					
S-B-ML	200.60†	0.30	14.59	[200.15, 201.06]	94.48
L-B-ML	196.20†	-1.90	14.77	[195.75, 196.64]	93.57*
Laplace-ML	200.36	0.18	14.53	[199.91, 200.81]	94.43
ZERO-REML	200.64†	0.32	14.63	[200.19, 201.09]	95.53
EBLUP-REML	196.34†	-1.83	14.75	[195.90, 196.78]	93.66*
FELA-EM-REML	200.65†	0.32	14.57	[200.20, 201.10]	94.63
$\beta_2 = 700$					
S-B-ML	696.46†	-0.51	47.64	[694.99, 697.93]	94.91
L-B-ML	682.62†	-2.48	46.43	[681.29, 683.95]	92.27*
Laplace-ML	703.92†	0.56	47.18	[702.46, 705.37]	95.40
ZERO-REML	697.18†	-0.40	47.24	[695.72, 698.64]	95.66
EBLUP-REML	683.06†	-2.42	46.44	[681.72, 684.40]	92.20*
FELA-EM-REML	702.75†	0.39	47.23	[701.29, 704.21]	95.25
$\beta_3 = 350$					
S-B-ML	360.48†	3.00	26.61	[359.72, 361.23]	96.49*
L-B-ML	335.12†	-4.25	25.85	[334.47, 335.78]	90.25*
Laplace-ML	353.14†	0.90	24.02	[352.40, 353.88]	96.50*
ZERO-REML	360.51†	3.00	27.39	[359.72, 361.29]	96.78*
EBLUP-REML	335.47†	-4.15	25.82	[334.81, 336.13]	89.60*
FELA-EM-REML	351.96†	0.56	23.97	[351.22, 352.70]	96.02

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

* The %CVR is outside the interval (93.97, 96.03).

Tables 3.3 and 3.4 contain the simulation results for the random-effects, the variance-covariance components in the logistic model for small and large D and σ^2, respectively. When D and σ^2 are small, the three REML approximations (FELA-EM-REML, ZERO-REML, and EBLUP-REML) all give unbiased estimates while the three ML approximations (Laplace-ML, S-B-ML, and L-B-ML) can significantly underestimate the estimators by up to 9.27%, i.e., L-B-ML for estimating D_{22}. For the variance component associated with the random effects that enter the nonlinear mixed model linearly D_{11}, the three ML approximations significantly underestimate the estimator by more than 6% while the three REML approximations give unbiased estimates with a maximum value of %Bias 0.79% for ZERO-REML. The results for D_{22}, which is the variance component associated with the random effects that enter the model nonlinearly, are similar to those for D_{11}. The three ML approximations again significantly underestimate the estimator and the %Bias increases to about 9%. The three REML approximations give unbiased estimates with a maximum absolute value of %Bias 0.44% for EBLUP-REML. For the covariance component D_{21} all six approximations give unbiased estimates. The relatively large %Bias (greater than 1%) indicates that the covariance component D_{21} is estimated with less accuracy than the variance components of the random effects D_{11} and D_{22}. The subject specific variance σ^2 is estimated with more relative accuracy than the elements of the variance-covariance components of the random effects (D_{11}, D_{21}, and D_{22}). This can be seen from the small %Bias for all six approximations (less than 1%). The possible reason is that the estimate of σ^2 is determined by the total number of observations, while the estimates of D are determined by the number of subjects. Although the 95% confidence intervals show that the three ML approximations provide significantly

negative biased estimates for σ^2, the %Bias are small with a maximum value 0.82% for Laplace-ML approximation. The three REML approximations again give unbiased estimates. When D and σ^2 increase, the only approximation method which gives unbiased estimates for all variance-covariance components is FELA-EM-REML. For D_{11}, the three ML approximations significantly underestimate the parameter by more than 5%. The two REML version linearization methods behave differently. The ZERO-REML significantly overestimates D_{11} by 1.98% while the EBLUP-REML gives a significantly negative biased estimate with %Bias equal to 1.33%. For D_{22}, both FELA-EM-REML and S-B-ML approximations give unbiased estimates with %Bias less than 1%. The ZERO-REML approximation again significantly overestimates D_{22} by up to 8.57% while the other three approximations (Laplace-ML, L-B-ML, and EBLUP-REML) significantly underestimate the estimator with a maximum value of %Bias up to 15.4% for L-B-ML. The %Bias for EBLUP-REML (6.26%) is similar to that for Laplace-ML (6.23%), which indicates that the REML eblup-expansion approximation cannot correct the bias for D_{22} in this case. For covariance component D_{21}, the three ML approximations and the two REML linearization methods all give significantly positive biased estimates with a maximum value of %Bias up to 18.64% for S-B-ML. Although the FELA-EM-REML approximation gives an unbiased estimate for D_{21}, the large %Bias (4.48%) again indicates that the covariance component is estimated with less accuracy than the variance components. The subject specific variance σ^2 is estimated with more relative accuracy than the variance-covariance components D. The %Bias is significant for all but the FELA-EM-REML approximation although the absolute value is small for all six approximations with a maximum absolute value 0.81 for Laplace-ML. The RMSE's for

each of the variance-covariance components D and σ^2 again are relatively similar for all six approximations considered no matter the size of D and σ^2.

Table 3.3: Simulation results for the random-effects variation in the logistic model for small variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11} = 100$				
S-B-ML	93.67†	-6.33	38.42	[92.50, 94.85]
L-B-ML	93.56†	-6.44	38.36	[92.39, 94.73]
Laplace-ML	93.69†	-6.31	38.41	[92.51, 94.86]
ZERO-REML	100.79	0.79	40.60	[99.53, 102.05]
EBLUP-REML	100.65	0.65	40.54	[99.40, 101.91]
FELA-EM-REML	100.76	0.76	40.60	[99.51, 102.02]
$D_{21} = -50$				
S-B-ML	-47.74	4.51	76.45	[-50.11, -45.37]
L-B-ML	-47.77	4.45	76.26	[-50.14, -45.41]
Laplace-ML	-47.80	4.40	76.48	[-50.17, -45.43]
ZERO-REML	-49.08	1.83	81.88	[-51.62, -46.55]
EBLUP-REML	-49.31	1.37	81.86	[-51.85, -46.78]
FELA-EM-REML	-49.39	1.21	81.95	[-51.93, -46.85]
$D_{22} = 625$				
S-B-ML	569.60†	-8.86	318.86	[559.87, 579.33]
L-B-ML	567.06†	-9.27	315.14	[557.46, 576.67]
Laplace-ML	569.41†	-8.89	316.28	[559.76, 579.06]
ZERO-REML	625.91	0.15	335.36	[615.51, 636.30]
EBLUP-REML	622.24	-0.44	333.20	[611.91, 632.56]
FELA-EM-REML	624.87	-0.02	333.97	[614.52, 635.22]
$\sigma^2 = 10$				
S-B-ML	9.92†	-0.76	1.28	[9.88, 9.96]
L-B-ML	9.92†	-0.81	1.27	[9.88, 9.96]
Laplace-ML	9.92†	-0.82	1.28	[9.88, 9.96]
ZERO-REML	10.01	0.08	1.28	[9.97, 10.05]
EBLUP-REML	10.01	0.08	1.29	[9.97, 10.05]
FELA-EM-REML	10.00	0.02	1.28	[9.96, 10.04]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

Table 3.4: Simulation results for the random-effects variation in the logistic model for large variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11} = 2500$				
S-B-ML	2370.64†	-5.17	981.84	[2340.48, 2400.81]
L-B-ML	2290.59†	-8.38	957.44	[2261.63, 2319.55]
Laplace-ML	2360.48†	-5.58	981.46	[2330.37, 2390.59]
ZERO-REML	2549.58†	1.98	1043.38	[2517.28, 2581.89]
EBLUP-REML	2466.79†	-1.33	1003.57	[2435.70, 2497.87]
FELA-EM-REML	2513.21	0.53	1039.63	[2480.99, 2545.43]
$D_{21} = -1250$				
S-B-ML	-1017.00†	18.64	2011.49	[-1078.93, -955.08]
L-B-ML	-1136.47†	9.08	2011.24	[-1198.71, -1074.23]
Laplace-ML	-1176.23†	5.90	2151.18	[-1242.86, -1109.60]
ZERO-REML	-1046.67†	16.27	2134.69	[-1112.53, -980.81]
EBLUP-REML	-1168.19†	6.54	2165.45	[-1235.26, -1101.12]
FELA-EM-REML	-1305.98	-4.48	2315.28	[-1377.72, -1234.24]
$D_{22} = 15625$				
S-B-ML	15608.36	-0.11	9824.41	[15303.87, 15912.86]
L-B-ML	13219.50†	-15.40	8091.66	[12980.05, 13458.95]
Laplace-ML	14652.17†	-6.23	8835.86	[14379.98, 14924.36]
ZERO-REML	16964.57†	8.57	10448.52	[16643.41, 17285.74]
EBLUP-REML	14646.60†	-6.26	8450.22	[14386.46, 14906.74]
FELA-EM-REML	15695.81	0.45	9363.32	[15405.61, 15986.00]
$\sigma^2 = 250$				
S-B-ML	248.44†	-0.62	32.00	[247.45, 249.43]
L-B-ML	248.95†	-0.42	32.05	[247.96, 249.94]
Laplace-ML	247.98†	-0.81	32.00	[246.99, 248.97]
ZERO-REML	251.02†	0.41	32.24	[250.03, 252.02]
EBLUP-REML	251.04†	0.42	32.40	[250.04, 252.05]
FELA-EM-REML	249.81	-0.07	32.40	[248.81, 250.82]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

3.4.2 First-order compartment model

A 3-parameter first-order compartment model with two random effects used by Pinheiro and Bates (1995) was used to generate the data. The values of the fixed-effects parameters, the covariate, and the dose were also similar to those used by Pinheiro and Bates (1995). The nonlinear mixed model is given by

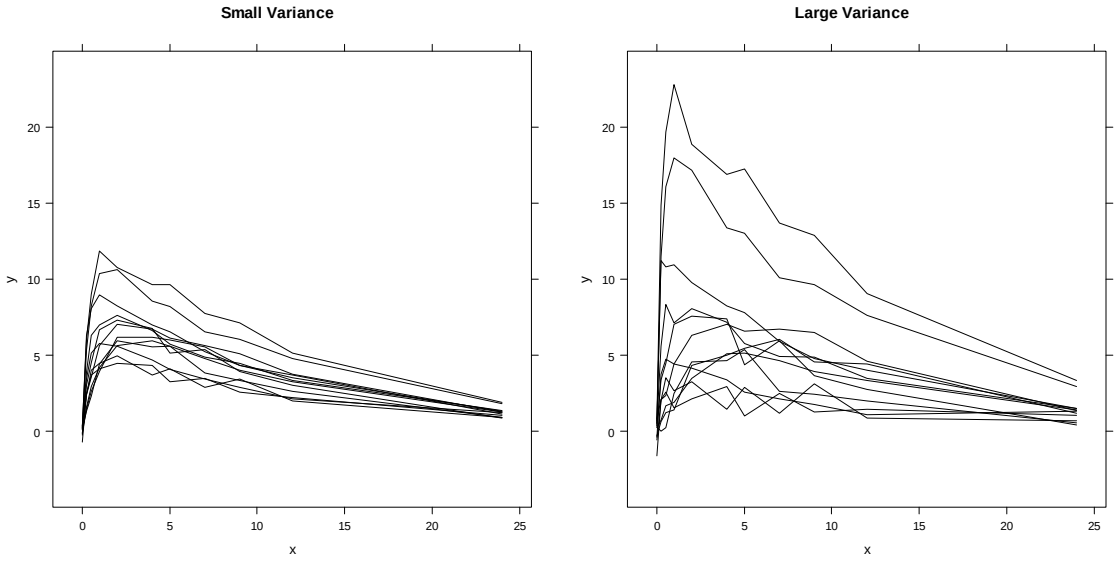
$$y_{ij} = \frac{\text{Dose} \cdot \exp[-(\beta_1 + b_{i1}) + (\beta_2 + b_{i2}) + \beta_3]}{\exp(\beta_2 + b_{i2}) - \exp(\beta_3)} \{\exp[-\exp(\beta_3)x_{ij}]$$

$$- \exp[-\exp(\beta_2 + b_{i2})x_{ij}]\} + \varepsilon_{ij}, \qquad i = 1,...,M, \quad j = 1,...,n_i,$$

(3.4.2)

where $\boldsymbol{b}_i = (b_{i1}, b_{i2})^T$ are i.i.d. $N(0, \boldsymbol{D})$. The ε_{ij} are i.i.d. $N(0, \sigma^2)$ and independent of $\boldsymbol{b}_i$. We use Dose =5, $M = 12$, $n_i = 11$ for $i = 1,...,M$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^T = (-3, 0.5, -2.5)^T$, and $\boldsymbol{x}_i = (0, 0.25, 0.5, 1, 2, 4, 5, 7, 9, 12, 24)^T$. Again, two situations based on the covariance matrix are considered, denoted as small variance and large variance, respectively. For small variance, the settings are similar to those used by Pinheiro and Bates (1995), $\boldsymbol{D} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.2 \end{bmatrix}$ and $\sigma^2 = 0.1$. For large variance, the values for $\boldsymbol{D}$ are similar to those used by Wolfinger and Lin (1997); we multiply each element of $\boldsymbol{D}$ and σ^2 for the small variance settings by 5, that is, $\boldsymbol{D} = \begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix}$ and $\sigma^2 = 0.5$. Figure 3.2 shows sample profiles for each of the two settings of variance and covariance parameters.

Figure 3.2: Example of simulated first-order compartment curves for small and large variance and covariance parameters



Tables 3.5 and 3.6 summarize the simulation results for the fixed-effects estimates in the first-order compartment model for small and large variance-covariance components D and σ^2, respectively. When D and σ^2 are small, FELA-EM-REML, Laplace-ML, and the two eblup-expansion linearization methods (L-B-ML and EBLUP-REML) provide more accurate estimates for the fixed-effects parameters than the two zero-expansion approximations (S-B-ML and ZERO-REML). For β_1, the two zero-expansion approximations significantly underestimate the estimator with %Bias by about 1.5% while the other four approximations show very little bias with a maximum absolute value of %Bias 0.11% for Laplace-ML and FELA-EM-REML. For β_2, the two zero-expansion approximations significantly overestimate the estimator by 2.85% and 7.08%, respectively, while the other four approximations all give unbiased estimates with a maximum absolute value of %Bias 0.61% for L-B-ML method. For β_3, which is the only non-random coefficient, the results are similar to those for β_1. Both S-B-ML and ZERO-

REML approximations again significantly underestimate the estimator with %Bias by about 1.65% while the other four approximations show very little bias with a maximum absolute value of %Bias 0.07% for the two eblup-expansion approximations (L-B-ML and EBLUP-REML). When D and σ^2 increase, only FELA-EM-REML and Laplace-ML approximations give unbiased or close-to-unbiased (%Bias < 1%) estimates for all three fixed effects. For β_1, the two zero-expansion approximations significantly underestimate the estimator with %Bias by 5.7% and 6.35%, respectively, while the other four approximations show very little negative bias with a maximum absolute value of %Bias 0.31% for L-B-ML. For β_2, the two zero-expansion approximations greatly overestimate the estimator by 12.78% and 32.82%, respectively, while the two eblup-expansion approximations significantly underestimate the estimator by about 7%. The two approximations which give unbiased estimates for β_2 are FELA-EM-REML and Laplace-ML. For the non-random coefficient β_3, the results again are similar to those for β_1. Both S-B-ML and ZERO-REML approximations significantly underestimate the estimator with %Bias by about 6% while the other four approximations provide very slight bias with a maximum absolute value of %Bias 0.28% for EBLUP-REML. The observed 95% confidence interval coverages are similar for both small and large variance situations. For β_1, the two zero-expansion linearization methods (S-B-ML and ZERO-REML) give considerable lower coverages while the other four approximations (L-B-ML, Laplace-ML, EBLUP-REML, and FELA-EM-REML) attain their nominal values. For β_2, both EBLUP-REML and FELA-EM-REML approximations give coverages similar to their nominal values while the coverage rates decrease significantly from the nominal ones for the other four methods (S-B-ML, L-B-ML, Laplace-ML, and ZERO-REML),

with the lowest coverage of 52.61% for S-B-ML for large D and σ^2. For β_3, the only nor-random coefficient and entering the model nonlinearly, the coverage rates obtained from all six approximations are significantly different from the nominal value. While the two zero-expansion linearization methods (S-B-ML and ZERO-REML) provide lower coverages compared with the nominal value, the coverages tend to increase from the nominal one for the other four approximations (L-B-ML, Laplace-ML, EBLUP-REML, and FELA-EM-REML). As in the logistic model analysis, we observe that the RMSE's for each of the three fixed effects are relatively similar for all six approximations considered no matter the size of the variance-covariance components D and σ^2.

Table 3.5: Simulation results for the fixed effects in the first-order compartment model for small variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI	%CVR
$\beta_1 = -3$					
S-B-ML	-3.0437†	-1.46	0.0818	[-3.0459, -3.0416]	89.92*
L-B-ML	-3.0026†	-0.09	0.0671	[-3.0047, -3.0006]	94.28
Laplace-ML	-3.0034†	-0.11	0.0670	[-3.0054, -3.0013]	94.35
ZERO-REML	-3.0449†	-1.50	0.0818	[-3.0470, -3.0428]	92.23*
EBLUP-REML	-3.0026†	-0.09	0.0671	[-3.0046, -3.0005]	95.17
FELA-EM-REML	-3.0033†	-0.11	0.0671	[-3.0054, -3.0012]	95.10
$\beta_2 = 0.5$					
S-B-ML	0.5143†	2.85	0.1479	[0.5097, 0.5188]	66.75*
L-B-ML	0.4969	-0.61	0.1360	[0.4927, 0.5012]	93.08*
Laplace-ML	0.5006	0.12	0.1362	[0.4964, 0.5048]	93.23*
ZERO-REML	0.5354†	7.08	0.1401	[0.5312, 0.5396]	93.40*
EBLUP-REML	0.4972	-0.57	0.1360	[0.4929, 0.5014]	94.10
FELA-EM-REML	0.5007	0.15	0.1363	[0.4965, 0.5050]	94.07
$\beta_3 = -2.5$					
S-B-ML	-2.5414†	-1.65	0.0522	[-2.5423, -2.5404]	76.73*
L-B-ML	-2.4982†	0.07	0.0267	[-2.4991, -2.4974]	97.05*
Laplace-ML	-2.5011†	-0.04	0.0267	[-2.5019, -2.5003]	97.13*
ZERO-REML	-2.5421†	-1.68	0.0532	[-2.5431, -2.5411]	76.98*
EBLUP-REML	-2.4982†	0.07	0.0267	[-2.4991, -2.4974]	97.25*
FELA-EM-REML	-2.5012†	-0.05	0.0267	[-2.5020, -2.5003]	97.22*

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

* The %CVR is outside the interval (93.97, 96.03).

Table 3.6: Simulation results for the fixed effects in the first-order compartment model for large variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI	%CVR
$\beta_1 = -3$					
S-B-ML	-3.1711†	-5.70	0.2384	[-3.1762, -3.1660]	79.57*
L-B-ML	-3.0092†	-0.31	0.1492	[-3.0138, -3.0046]	94.37
Laplace-ML	-3.0087†	-0.29	0.1490	[-3.0133, -3.0041]	94.43
ZERO-REML	-3.1905†	-6.35	0.2492	[-3.1955, -3.1855]	82.96*
EBLUP-REML	-3.0086†	-0.29	0.1493	[-3.0132, -3.0040]	95.07
FELA-EM-REML	-3.0086†	-0.29	0.1492	[-3.0132, -3.0039]	95.44
$\beta_2 = 0.5$					
S-B-ML	0.5639†	12.78	0.3074	[0.5546, 0.5732]	52.61*
L-B-ML	0.4636†	-7.27	0.2989	[0.4544, 0.4728]	93.29*
Laplace-ML	0.4984	-0.33	0.3093	[0.4888, 0.5080]	93.43*
ZERO-REML	0.6641†	32.82	0.3408	[0.6549, 0.6734]	88.68*
EBLUP-REML	0.4652†	-6.95	0.2998	[0.4560, 0.4745]	94.12
FELA-EM-REML	0.5038	0.76	0.3127	[0.4941, 0.5135]	94.09
$\beta_3 = -2.5$					
S-B-ML	-2.6486†	-5.94	0.1748	[-2.6515, -2.6458]	47.01*
L-B-ML	-2.4932†	0.27	0.0541	[-2.4949, -2.4916]	97.05*
Laplace-ML	-2.5048†	-0.19	0.0531	[-2.5064, -2.5031]	97.13*
ZERO-REML	-2.6698†	-6.79	0.1938	[-2.6727, -2.6669]	46.45*
EBLUP-REML	-2.4931†	0.28	0.0533	[-2.4947, -2.4914]	97.06*
FELA-EM-REML	-2.5049†	-0.20	0.0613	[-2.5068, -2.5030]	97.12*

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

* The %CVR is outside the interval (93.97, 96.03).

Tables 3.7 and 3.8 show the simulation results for the variance-covariance components in the first-order compartment model for small and large D and σ^2, respectively. When D and σ^2 are small, both FELA-EM-REML and EBLUP-REML approximations give unbiased estimates for all the variance-covariance parameters. For D_{11}, the three ML approximations (S-B-ML, L-B-ML, and Laplace-ML) significantly underestimate the estimator by more than 6% while the ZERO-REML approximation gives a significant positive biased estimate (1.91%). For D_{22}, the three ML approximations and the ZERO-REML approximation all provide significant negative biased estimates with a maximum value of %Bias 11.08% for S-B-ML. For the covariance component D_{21}, the %Bias cannot be obtained since the true value of D_{21} equals 0. However, by investigating the 95% confidence intervals of the covariance parameters, we found that the two zero-expansion approximations (S-B-ML and ZERO-REML) significantly underestimate the estimator, while L-B-ML and Laplace-ML approximations also give unbiased estimates. For the subject specific variance σ^2, the two zero-expansion approximations greatly overestimate the estimator by 13.79% and 17.32%, respectively, while L-B-ML and Laplace-ML both give close-to-unbiased estimates with a maximum value of %Bias 0.94% for Laplace-ML. When both D and σ^2 are large, the only approximation method which gives unbiased or little biased estimates for all variance-covariance components is FELA-EM-REML. For D_{11}, both L-B-ML and Laplace-ML approximations significantly underestimate the parameter by more than 7%, while the two zero-expansion approximations significantly overestimate the parameter by 2.72% and 9.29%, respectively. Both FELA-EM-REML and EBLUP-REML give unbiased estimates for D_{11}. For D_{22}, the only approximation method which gives an unbiased estimate is FELA-

EM-REML, while the other five approximations all give significantly negative biased estimates. The two zero-expansion and L-B-ML approximations greatly underestimate D_{22} by 25.83%, 24.94%, and 14.85%, respectively, while Laplace-ML and EBLUP-REML give moderately biased estimates with %Bias 8.2% and 6.24%, respectively. For the covariance component D_{21}, the two zero-expansion approximations significantly underestimate the estimator while FELA-EM-REML shows very little positive bias. The other three approximations all give unbiased estimates for D_{21}. For the subject specific variance σ^2, the two zero-expansion approximations greatly overestimate the estimator by 39.06% and 48%, respectively, while the other four approximations provide more accurate estimates – either unbiased (L-B-ML and FELA-EM-REML) or slightly biased (Laplace-ML and EBLUP-REML) with a maximum absolute value of %Bias 1.28% for Laplace-ML. The RMSE's for each of the variance-covariance components D and σ^2 are relatively similar for all six approximations considered no matter the size of D and σ^2.

Table 3.7: Simulation results for the random-effects variation in the first-order compartment model for small variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11} = 0.05$				
S-B-ML	0.0469†	-6.13	0.0227	[0.0462, 0.0476]
L-B-ML	0.0457†	-8.51	0.0206	[0.0451, 0.0464]
Laplace-ML	0.0459†	-8.14	0.0207	[0.0453, 0.0466]
ZERO-REML	0.0510†	1.91	0.0242	[0.0502, 0.0517]
EBLUP-REML	0.0500	-0.07	0.0220	[0.0493, 0.0506]
FELA-EM-REML	0.0502	0.34	0.0221	[0.0495, 0.0509]
$D_{21} = 0$				
S-B-ML	-0.0022†	N/A	0.0311	[-0.0031, -0.0012]
L-B-ML	0.0004	N/A	0.0281	[-0.0005, 0.0013]
Laplace-ML	0.0005	N/A	0.0282	[-0.0004, 0.0014]
ZERO-REML	-0.0015†	N/A	0.0325	[-0.0025, -0.0005]
EBLUP-REML	0.0005	N/A	0.0307	[-0.0004, 0.0015]
FELA-EM-REML	0.0007	N/A	0.0309	[-0.0002, 0.0017]
$D_{22} = 0.2$				
S-B-ML	0.1778†	-11.08	0.0832	[0.1754, 0.1803]
L-B-ML	0.1822†	-8.89	0.0850	[0.1796, 0.1848]
Laplace-ML	0.1831†	-8.47	0.0855	[0.1805, 0.1857]
ZERO-REML	0.1853†	-7.35	0.0836	[0.1828, 0.1879]
EBLUP-REML	0.1995	-0.26	0.0908	[0.1967, 0.2023]
FELA-EM-REML	0.2007	0.36	0.0917	[0.1979, 0.2036]
$\sigma^2 = 0.1$				
S-B-ML	0.1138†	13.79	0.0243	[0.1132, 0.1144]
L-B-ML	0.0991†	-0.87	0.0135	[0.0987, 0.0995]
Laplace-ML	0.0991†	-0.94	0.0135	[0.0986, 0.0995]
ZERO-REML	0.1173†	17.32	0.0278	[0.1166, 0.1180]
EBLUP-REML	0.1001	0.05	0.0136	[0.0996, 0.1005]
FELA-EM-REML	0.1000	-0.04	0.0135	[0.0995, 0.1004]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

Table 3.8: Simulation results for the random-effects variation in the first-order compartment model for large variance and covariance parameters

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11} = 0.25$				
S-B-ML	0.2568†	2.72	0.1635	[0.2517, 0.2619]
L-B-ML	0.2252†	-9.93	0.1042	[0.2220, 0.2283]
Laplace-ML	0.2310†	-7.59	0.1063	[0.2278, 0.2343]
ZERO-REML	0.2732†	9.29	0.1622	[0.2682, 0.2782]
EBLUP-REML	0.2473	-1.09	0.1277	[0.2433, 0.2512]
FELA-EM-REML	0.2516	0.65	0.1136	[0.2481, 0.2551]
$D_{21} = 0$				
S-B-ML	-0.0480†	N/A	0.2007	[-0.0541, -0.0420]
L-B-ML	-0.0026	N/A	0.1432	[-0.0070, 0.0018]
Laplace-ML	0.0037	N/A	0.1535	[-0.0011, 0.0084]
ZERO-REML	-0.0314†	N/A	0.1996	[-0.0375, -0.0252]
EBLUP-REML	-0.0007	N/A	0.1694	[-0.0060, 0.0046]
FELA-EM-REML	0.0074†	N/A	0.1667	[0.0022, 0.0126]
$D_{22} = 1$				
S-B-ML	0.7417†	-25.83	0.4533	[0.7301, 0.7532]
L-B-ML	0.8515†	-14.85	0.4234	[0.8392, 0.8638]
Laplace-ML	0.9180†	-8.20	0.4754	[0.9035, 0.9325]
ZERO-REML	0.7506†	-24.94	0.4448	[0.7392, 0.7620]
EBLUP-REML	0.9376†	-6.24	0.4440	[0.9240, 0.9512]
FELA-EM-REML	1.0085	0.85	0.4952	[0.9932, 1.0239]
$\sigma^2 = 0.5$				
S-B-ML	0.6953†	39.06	0.2625	[0.6899, 0.7008]
L-B-ML	0.4982	-0.36	0.0716	[0.4960, 0.5004]
Laplace-ML	0.4936†	-1.28	0.0672	[0.4915, 0.4957]
ZERO-REML	0.7400†	48.00	0.3196	[0.7335, 0.7465]
EBLUP-REML	0.5031†	0.62	0.0855	[0.5004, 0.5057]
FELA-EM-REML	0.4984	-0.31	0.0705	[0.4963, 0.5006]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

3.5 Discussion

We have developed a new EM algorithm, a fully exponential Laplace approximation EM algorithm (FELA-EM), for REML estimation in parametric nonlinear mixed effects models, where we assume that both random effects and error are normally distributed. The new algorithm is generally as accurate as the direct approximations of marginal likelihood estimation for nonlinear mixed models based on third-order expansions while preserving the computational simplicity of those methods based on second-order expansions (i.e., Laplace approximation). The proposed algorithm gives an error of second order $O(1/n^2)$ compared with the first-order error $O(1/n)$ from the standard Laplace method for estimating variance-covariance components. The computing time of the FELA-EM algorithm is about 4-5 times longer than that of the Laplace approximation and 20-25 times than that of the linearization methods (i.e., L-B-ML) while the computing time of numerical integration and Monte Carlo methods can be more than 1000 times longer than that of the linearization methods. Thus, the FELA-EM algorithm is computationally much more efficient than numerical integration and Monte Carlo methods. Simulation studies for both the logistic model and the compartment model considered in this paper are performed for two situations-either small or large variability in both random effects and residual error. This is a generalization of conditions used by Pinheiro and Bates (1995) and Wolfinger and Lin (1997). Pinheiro and Bates (1995) consider the small variability in both random effects and residual error, while Wolfinger and Lin (1997) use both small and large variability for random effects but consider only small variability in residual error.

The results of Section 3.4 indicate that the proposed FELA-EM-REML algorithm for nonlinear mixed effects models gives accurate and reliable REML estimate for both fixed effects and variance-covariance components. It produces either unbiased or close-to-unbiased (%Bias < 1%) estimates for both the fixed effects and the variance-covariance components and gives 95% confidence interval coverages similar to the nominal value for all the fixed effects. For the fixed-effects estimation, the estimates obtained from FELA-EM-REML approximation are similar to those from the Laplace-ML approximation, but are more accurate than those from the four linearization methods. Both FELA-EM-REML and Laplace-ML approximations give unbiased or close-to-unbiased estimates with 95% confidence interval coverages similar to the nominal value for all fixed effects no matter the size of D and σ^2 while the four linearization methods frequently generate significantly biased estimates for large D and σ^2. The coverage rates obtained from the four linearization methods can be much lower than the nominal 95% level with the lowest coverage of about 47% for the two zero-expansion approximations for estimating β_3 in the compartment model when both of D and σ^2 are large. We also observe that the two eblup-expansion approximations generally give smaller bias than the two zero-expansion approximations and the two zero-expansion approximations can give estimates with %Bias as high as 33%. The RMSE's for each of the fixed effects are relatively similar for all approximations considered no matter the size of D and σ^2.

For the variance-covariance estimation, our proposed FELA-EM-REML algorithm is the only approximation method that always gives unbiased estimates for variance and

covariance parameters. Although the EBLUP-REML approximation also generates unbiased estimates for variance-covariance components for small D and σ^2, it generally gives significantly biased estimates for large D and σ^2. The three ML approximations (Laplace-ML, S-B-ML, and L-B-ML) generally underestimate the variance parameters no matter the size of D and σ^2. For covariance parameters, the performance of the ML approximations does have a consistent pattern. They can generate either unbiased or significantly biased estimates. The ZERO-REML approximation generally gives more accurate estimates than the three ML approximations for small D and σ^2. However, it can generate very poor estimates with %Bias up to 48% when both D and σ^2 are large. The RMSE's for each of the variance-covariance components are relatively similar for all approximations considered no matter the size of D and σ^2.

3.6 Conclusions

The proposed FELA-EM algorithm for nonlinear mixed effects models gives accurate and reliable REML estimations (either unbiased or close-to-unbiased) for both fixed effects and variance-covariance components. For the fixed-effects estimation, both the FELA-EM algorithm and the Laplace approximation gives either unbiased or close-to-unbiased estimates with 95% confidence interval coverages similar to the nominal value for all fixed effects no matter the size of D and σ^2 while the four linearization methods can frequently generate significantly biased estimates for large D and σ^2. For the variance-covariance estimation, the EBLUP-REML approximation gives unbiased estimates for small D and σ^2 but generates significantly biased estimates for large D and σ^2. The three ML approximations frequently underestimate the variance parameters

for both small and large D and σ^2, and their performance for covariance is uncertain (either unbiased or significantly biased). The ZERO-REML approximation is more accurate than the three ML approximations for small D and σ^2 but it can produce generate very poor estimates when both D and σ^2 are large.

3.7 Summary

In this paper we have developed a new EM algorithm to obtain REML estimates of variance-covariance components as well as both the fixed and random effects parameters in nonlinear mixed effects models. This new approach, called fully exponential Laplace approximation EM algorithm (FELA-EM-REML), can be considered as an improvement of the Laplace approximation which preserves the computational efficiency and improves the accuracy of the parameter estimation of Laplace approximation. This new approach first applies the standard EM algorithm to nonlinear mixed effects models and then uses the fully exponential Laplace method to approximate the conditional expectations of the complete data sufficient statistics in the E-step. Four simulation studies (two models each with small and large variance-covariance parameters) were conducted to evaluate the accuracy of the new approach and compare it with the ML Laplace approximation and four linearization methods. Of all approximations considered in the paper, the proposed FELA-EM-REML algorithm is the only one that gives unbiased or close-to-unbiased (%Bias $< 1\%$) estimates for all the fixed effects and variance-covariance components with 95% confidence interval coverages similar to the nominal value for all the fixed effects.

3.8 References

Beal, S.L., Sheiner, L.B., 1992. NONMEM Users Guide. University of California, San Francisco, NONMEM Project Group.

Breslow, N.E., Lin, X., 1995. Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika* 82, 81-91.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society*, Series B 39, 1-38.

Ge, Z., Bickel, J.P., Rice, A.J., 2004. An approximate likelihood approach to nonlinear mixed effects models via spline approximation. *Computational Statistics & Data Analysis* 46, 747-776.

Insightful Corporation, 2007. S-PLUS 8 for Windows Guide to Statistics. Seattle, WA, Insightful Corporation.

Jamshidian, M., Jennrich, R.I., 1997. Acceleration of the EM Algorithm by using Quasi-Newton Methods. *Journal of the Royal Statistical Society*, Series B 59, 569-587.

Kauermann, G., Xu, R., Vaida, F., 2008. Stacked Laplace-EM algorithm for duration models with time-varying and random effects. *Computational Statistics & Data Analysis* 52, 2514-2528.

Kuhn, E., Lavielle, M., 2005. Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics & Data Analysis* 49, 1020-1038.

Lindstrom, M.J., Bates, D.M., 1990. Nonlinear mixed effects models for repeated measures data. *Biometrics* 46, 673-687.

Noh, M., Lee, Y., 2008. Hierarchical-likelihood approach for nonlinear mixed-effects models. *Computational Statistics & Data Analysis* 52, 3517-3527.

Pinheiro, J.C., Bates, D.M., 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4, 12-35.

Pinheiro, J.C., Bates, D.M., 2000. Mixed-Effects Models in S and S-PLUS. Springer, New York.

R Development Core Team, 2009. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org

Ratkowsky, D.A., 1983. Nonlinear regression modeling: a unified practical approach. Marcel Dekker, NY.

SAS Institute, 2004. SAS OnlineDoc, Version 9.1.3. SAS Institute Inc., Cary.

Sheiner, L.B., Beal, S.L., 1980. Evaluation of methods for estimating population pharmokinetics. I. Michaelis-Menton Model: Routine clinical pharacokinetics data. *Journal of Pharmacokinetics and Biopharmaceutics* 8, 673-687.

Shun, Z., McCullagh, P., 1995. Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society*, Series B 57, 749-760.

Tierney, L., Kadane, J.B., 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81, 82-86.

Tierney, L., Kass, R.E., Kadane, J.B., 1989. Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association* 84, 710-716.

Vonesh, E.F., 1996. A note on the use of Laplace's approximation for nonlinear mixed-effects models. *Biometrika* 83, 447-452.

Vonesh, E.F., Wang, H., Nie, L., Majumdar, D., 2002. Conditional second-order generalized estimating equations for generalized linear and nonlinear mixed-effects models. *Journal of the American Statistical Association* 97, 271-283.

Wakefield, J.C., Smith, A.F.M., Racine-Poon, A., Gelfand, A.E., 1994. Bayesian analysis of linear and nonlinear population models by using the Gibbs sampler. *Applied Statistics* 43, 201-221.

Walker, S., 1996. An EM algorithm for nonlinear random effects models. *Biometrics* 52, 934-944.

Wang, J., 2007. EM algorithms for nonlinear mixed effects models. *Computational Statistics & Data Analysis* 51, 3244-3256.

Wolfinger, R., 1993. Laplace's approximation for nonlinear mixed models. *Biometrika* 80, 791-795.

Wolfinger, R.D., Lin, X., 1997. Two Taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis* 25, 465-490.

CHAPTER 4

EXTENSION OF THE FULLY EXPONENTIAL LAPLACE APPROXIMATION EM ALGORITHM FOR NONLINEAR MIXED MODELS WITH TWO LEVELS OF CROSSED RANDOM EFFECTS

4.0 Abstract

Multilevel nonlinear mixed effects models have received much interest in the statistical literature in recent years. These models are useful for analyzing data presenting multiple levels of grouping. Examples include pharmaceutical studies in which concentration measurements are taken over time for patients from different clinical centers after given a drug or field experiments where measurements are obtained weekly from plants that grow in different fields. The most successful methods used to estimate the parameters in multilevel nonlinear mixed models are called linearization methods, which are based on using a first-order Taylor series expansion to approximate the nonlinear model function and maximizing the likelihood corresponding to the resulting approximate model. Although linearization methods are popular and computationally simple, they can produce substantial bias in parameter estimation with limited number of observations per stratum and large variability of random effects. Recently, several more accurate estimation methods for single-level nonlinear mixed models have been extended for parameter estimation in multilevel nonlinear mixed models including the maximum likelihood (ML) version of Laplace approximation and the stochastic approximation version of EM algorithm. This paper focuses on parameter estimation of nonlinear mixed effects models with two levels of crossed grouping. The restricted maximum likelihood

estimates are obtained via the extension of the FELA-EM algorithm. Two simulation studies have been conducted to evaluate the accuracy of the approach and compare it with the ML version of Laplace approximation and different linearization methods. Of all approximations considered in the paper, FELA-EM algorithm is the only one that gives unbiased or close-to-unbiased (%Bias < 1%, Ratkowsky 1983) estimates for both the fixed effects and variance-covariance parameters with 95% confidence interval coverages similar to the nominal value for all the fixed effects.

Key Words: Nonlinear mixed models; linearization methods; Laplace approximation; EM algorithm; Fully exponential Laplace approximation; Crossed random effects; ML estimation; REML estimation.

4.1 Introduction

Nonlinear mixed effects models with single-lever grouping factors have been widely used in numerous fields, such as biology, agriculture, environment, medicine, and economics for analyzing repeated measure data. In recent years, multilevel nonlinear mixed effects models, including nonlinear mixed models with nested random effects, nonlinear mixed models with crossed random effects, and nonlinear mixed models with both nested and crossed random effects, have begun to receive much interest in the statistical literature (Pinheiro and Bates 2000; Davidian and Giltinan 2003; Millar 2004; Panhard and Samson 2009). These models are useful for analyzing data presenting multiple levels of grouping such as in pharmaceutical studies where concentration measurements are taken over time for patients from different clinical centers after given a drug and in field experiments where measurements are obtained weekly from plants that grow in different fields.

The most popular methods used to estimate the parameters in multilevel nonlinear mixed models are called linearization methods, which are based on using a first-order Taylor series expansion to approximate the nonlinear model function and maximize the likelihood corresponding to the resulting approximate model. Linearization methods for multilevel nonlinear mixed models are computationally simple and have been implemented in a number of software packages such as NONMEM (Beal and Sheiner 1992), the %nlinmix macro (Wolfinger and Lin 1997) in SAS (SAS Institute 2004), and the nlme function (Pinheiro and Bates 2000) in both S-PLUS (Insightful Corporation 2007) and R (R development core team 2009). The only popular program that can solve multilevel nonlinear mixed models with both crossed and nested random effects is

the %nlinmix macro in SAS and the other programs generally only deal with multilevel nonlinear mixed models with nested random effects. The SAS macro solves the nonlinear mixed model based on the following idea: 1) Take a first-order Taylor series of the model around the current estimates of the fixed effects β and some values of the random effects b. This yields an approximate model of the linear mixed model form. 2) Fit this model with a linear mixed model procedure (i.e., the Mixed Procedure in SAS). 3) Update the expansion loci. 4) Repeat the process until a convergence criterion is met. There are different ways to choose the expansion locus of the random effects and we focus on two of them: expansion around zero that is the expected value of the random effects, denoted as ZERO and expansion around the estimated best linear unbiased predictor, denoted as EBLUP. Several different methods for estimating parameters in linear mixed-effects models have also been proposed and we concentrate on two of them: maximum likelihood, ML (Hartley and Rao 1967) and restricted maximum likelihood, REML (Harville 1977). Although linearization methods are popular and computationally simple, they can produce substantial bias in parameter estimation with limited number of observations per subject and large variability in random effects (Vonesh 1996; Ge, Bickel, and Rice 2004).

This bias has motivated researchers to search for more accurate methods for single-level nonlinear mixed models to improve the estimation in multilevel nonlinear mixed models. The ML version of Laplace approximation for single-level nonlinear mixed models proposed by Pinheiro and Bates (1995) has been extended to nonlinear mixed models with both crossed and nested random effects by Bates, Maechler, and Dai (2008). It is

also possible to implement the adaptive Gaussian quadrature (AGQ) method for single-level nonlinear mixed models proposed by Pinheiro and Bates (1995) in the SAS procedure NLMIXED for nonlinear mixed models with two nested groupings. However, due to the large number of quadrature points, the AGQ method often requires very high computer resources and time. Moreover, in practice, the AGQ method frequently fails to converge (Jaffrezic, Meza, Lavielle, and Foulley 2006).

As an alternative to the direct approximate to the marginal likelihood of nonlinear mixed effects models, various versions of EM algorithms (Dempster, Laird, and Rubin 1977) have been proposed for the parameter estimation in single-level nonlinear mixed models including Monte Carlo EM-based algorithms (Walker 1996; Wang 2007) and stochastic approximation EM algorithm (Kuhn and Lavielle 2005). The stochastic approximation version of EM algorithm for single-level nonlinear mixed models has been successfully extended to nonlinear mixed models with two nested grouping factors by Panhard and Samson (2009) while none of the Monte Carlo EM-based algorithms are directly applicable to the case of multilevel nonlinear mixed models.

The objective of this paper is to extend the FELA-EM algorithm (proposed in chapter 3) to nonlinear mixed effects models with two levels of crossed grouping for obtaining REML estimates. In Section 4.2, we present a general formulation for nonlinear mixed models with two levels of crossed grouping. In section 4.3, we give both ML and REML versions of Laplace approximation to the two-level crossed nonlinear mixed models. The extension of the FELA-EM algorithm for single-level nonlinear mixed models to

multilevel nonlinear mixed models with two crossed groupings is introduced in Section 4.4. In Section 4.5, we evaluate the accuracy of the extended approach and compare it with the ML version of Laplace approximation and four different linearization methods given in Wolfinger and Lin (1997). We give our overall conclusions in Section 4.6 and summarize the paper in Section 4.7.

4.2 Model and likelihood

For a multilevel nonlinear mixed effects model with two crossed grouping factors, the model can be written as

$$y_{ijk} = f(\boldsymbol{\beta}, \boldsymbol{b}_i^{(1)}, \boldsymbol{b}_j^{(2)}) + \varepsilon_{ijk}, \quad i = 1,..., M_1, \quad j = 1,..., M_2, \quad k = 1,..., n_{ij}, \tag{4.2.1}$$

where y_{ijk} is the kth observation on the ith unit of grouping factor 1 and the jth unit of grouping factor 2, f is a nonlinear function, $\boldsymbol{\beta}$ is a p-dimensional vector of fixed effects, $\boldsymbol{b}_i^{(1)}$ is a q_1-dimensional random effects vector associated with the ith unit of grouping factor 1 and assumed i.i.d. $N(\mathbf{0}, \boldsymbol{D}^{(1)})$, $\boldsymbol{b}_j^{(2)}$ is a q_2-dimensional random effects vector associated with the jth unit of grouping factor 2 and assumed i.i.d. $N(\mathbf{0}, \boldsymbol{D}^{(2)})$, ε_{ijk} is the error and assumed i.i.d. $N(0, \sigma^2)$, M_1 and M_2 are the number of units in grouping factors 1 and 2, respectively, and n_{ij} is the number of observations on the ith unit of grouping factor 1 and the jth unit of grouping factor 2. It is further assumed that $\boldsymbol{b}_i^{(1)}$, $\boldsymbol{b}_j^{(2)}$, and ε_{ijk} are independent.

The joint density of $\boldsymbol{y}$, $\boldsymbol{b}^{(1)}$, and $\boldsymbol{b}^{(2)}$ of model (4.2.1) is $p(\boldsymbol{y},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{D}^{(1)},\boldsymbol{D}^{(2)})$, where $\boldsymbol{y}$ is observed, but the random coefficients $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$ are unobserved. Because the random effects are unobserved quantities, ML estimation for model (4.2.1) is based on the marginal density of the responses, $\boldsymbol{y}$, which is calculated as

$$p(\boldsymbol{y}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{D}^{(1)},\boldsymbol{D}^{(2)}) = \int\{p(\boldsymbol{y}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)})p(\boldsymbol{b}^{(1)}\mid\boldsymbol{D}^{(1)}) \times p(\boldsymbol{b}^{(2)}\mid\boldsymbol{D}^{(2)})\}d\boldsymbol{b}^{(1)}d\boldsymbol{b}^{(2)}, \tag{4.2.2}$$

where $p(\boldsymbol{y}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{D}^{(1)},\boldsymbol{D}^{(2)})$ is the marginal density of $\boldsymbol{y}$ for ML estimation, $p(\boldsymbol{y}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)})$ is the conditional density of $\boldsymbol{y}$ given the random effects $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$, and the marginal distributions of $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$ are $p(\boldsymbol{b}^{(1)}\mid\boldsymbol{D}^{(1)})$ and $p(\boldsymbol{b}^{(2)}\mid\boldsymbol{D}^{(2)})$, respectively.

Similarly, by assuming a flat prior for the fixed effects $\boldsymbol{\beta}$, REML estimation for variance-covariance components $\boldsymbol{D}^{(1)}$, $\boldsymbol{D}^{(2)}$, and σ^2 in model (4.2.1) can be obtained by integrating out both fixed effects and random effects from the joint density of $\boldsymbol{y}$, $\boldsymbol{\beta}$, $\boldsymbol{b}^{(1)}$, and $\boldsymbol{b}^{(2)}$, which is given by

$$p(\boldsymbol{y} \mid \sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)}) = \int p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \mid \sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})\, d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}$$

$$= \int \Big\{ p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)})\, p(\boldsymbol{b}^{(1)} \mid \boldsymbol{D}^{(1)}) \tag{4.2.3}$$

$$\times\, p(\boldsymbol{b}^{(2)} \mid \boldsymbol{D}^{(2)}) \Big\} d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)},$$

where $p(\boldsymbol{y} \mid \sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ is the marginal density of $\boldsymbol{y}$ for REML estimation, $p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)})$ is the conditional density of $\boldsymbol{y}$ given both fixed effects $\boldsymbol{\beta}$ and random effects $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$, and the marginal distributions of $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$ are $p(\boldsymbol{b}^{(1)} \mid \boldsymbol{D}^{(1)})$ and $p(\boldsymbol{b}^{(2)} \mid \boldsymbol{D}^{(2)})$, respectively. Because the model function f can be nonlinear in the random effects, the integrals in both (4.2.2) and (4.2.3) generally do not have a closed-form expression.

4.3 Laplace approximation to the likelihood

The Laplace approximation is a method for approximating integrals using local information about the integrand at its maximum. Therefore, it is most useful when the integrand is highly concentrated about its maximizing value. The Laplace approximation has been widely used in Bayesian inference to compute marginal posterior densities (Tierney and Kadane 1986; Leonard, Hsu, and Tsui 1989) and it converges to the correct value as the sample size goes to infinity. We first apply Laplace approximation to the marginal density of $\boldsymbol{y}$ for ML estimation given in (4.2.2). The details are as follows.

The integral (4.2.2) that we want to estimate can be further written as

$$p(\boldsymbol{y}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{D}^{(1)},\boldsymbol{D}^{(2)}) = (2\pi\sigma^2)^{\frac{-N}{2}}(2\pi)^{\frac{-(M_1q_1+M_2q_2)}{2}}\left|\boldsymbol{D}^{(1)}\right|^{-\frac{M_1}{2}} \times\left|\boldsymbol{D}^{(2)}\right|^{-\frac{M_2}{2}}\int\exp\left[\frac{-g(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\boldsymbol{b})}{2}\right]d\boldsymbol{b}^{(1)}d\boldsymbol{b}^{(2)}, \tag{4.3.1}$$

where $g(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\boldsymbol{b}) = \sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\frac{\left\|\boldsymbol{y}_{ij}-f(\boldsymbol{\beta},\boldsymbol{b}_i^{(1)},\boldsymbol{b}_j^{(2)})\right\|^2}{\sigma^2}+\sum_{i=1}^{M_1}\boldsymbol{b}_i^{(1)T}\boldsymbol{D}^{(1)-1}\boldsymbol{b}_i^{(1)}+\sum_{i=1}^{M_2}\boldsymbol{b}_j^{(2)T}\boldsymbol{D}^{(2)-1}\boldsymbol{b}_j^{(2)}$

and $N=\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}n_{ij}$. Applying Laplace approximation to the integral in (4.3.1), we get

$$p(\boldsymbol{y}\mid\boldsymbol{\beta},\sigma^2,\boldsymbol{D}^{(1)},\boldsymbol{D}^{(2)}) \approx (2\pi\sigma^2)^{\frac{-N}{2}}\left|\boldsymbol{D}^{(1)}\right|^{-\frac{M_1}{2}}\left|\boldsymbol{D}^{(2)}\right|^{-\frac{M_2}{2}} \times\exp\left[\frac{-g(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\hat{\boldsymbol{b}})}{2}\right]\left|\frac{g''(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\hat{\boldsymbol{b}})}{2}\right|^{\frac{-1}{2}}, \tag{4.3.2}$$

where $\hat{\boldsymbol{b}}$ minimizes the function $g(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\boldsymbol{b})$ and $g''(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\hat{\boldsymbol{b}})$ is the second derivative of $g(\boldsymbol{\beta},\sigma^2,\boldsymbol{D},\boldsymbol{y},\boldsymbol{b})$ with respect to the random effects $\boldsymbol{b}$ evaluated at $\boldsymbol{b}=\hat{\boldsymbol{b}}$.

Similarly, the Laplace approximation to the marginal density of $\boldsymbol{y}$ for REML estimation given in (4.2.3) can be obtained by

$$p(\boldsymbol{y} \mid \sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)}) \approx (2\pi)^{\frac{pN}{2}} (2\pi\sigma^2)^{\frac{-N}{2}} \left|\boldsymbol{D}^{(1)}\right|^{-\frac{M_1}{2}} \left|\boldsymbol{D}^{(2)}\right|^{-\frac{M_2}{2}} \times \exp\left[\frac{-g(\hat{\boldsymbol{\beta}}, \sigma^2, \boldsymbol{D}, \boldsymbol{y}, \hat{\boldsymbol{b}})}{2}\right] \left|\frac{g''(\hat{\boldsymbol{\beta}}, \sigma^2, \boldsymbol{D}, \boldsymbol{y}, \hat{\boldsymbol{b}})}{2}\right|^{\frac{-1}{2}}, \tag{4.3.3}$$

where pN is the number of fixed effects $\boldsymbol{\beta}$ in model (4.2.1), $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}$ jointly minimize the function $g(\boldsymbol{\beta}, \sigma^2, \boldsymbol{D}, \boldsymbol{y}, \boldsymbol{b})$ and $g''(\hat{\boldsymbol{\beta}}, \sigma^2, \boldsymbol{D}, \boldsymbol{y}, \hat{\boldsymbol{b}})$ is the second derivative of $g(\boldsymbol{\beta}, \sigma^2, \boldsymbol{D}, \boldsymbol{y}, \boldsymbol{b})$ with respect to the fixed effects $\boldsymbol{\beta}$ and the random effects $\boldsymbol{b}$ evaluated at $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$ and $\boldsymbol{b} = \hat{\boldsymbol{b}}$, respectively.

4.4 FELA-EM algorithm

EM algorithm (Dempster, Laird, and Rubin 1977) has been widely used for finding ML/REML estimates in incomplete data problems due to its simplicity and stability. It alternates between two steps, an expectation step (E-step) and a maximization step (M-step). Let $c(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis})$ be the complete data where $\boldsymbol{y}_{obs}$ and $\boldsymbol{y}_{mis}$ represent the observed and missing data, respectively, and $\boldsymbol{\theta}$ be the vector of parameters to be estimated. On the $(K+1)$th iteration, the E-step generally computes the expectation of the complete data log-likelihood $\ell(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis}; \boldsymbol{\theta})$ conditional on the observed data $\boldsymbol{y}_{obs}$ at the current estimates of the parameters $\boldsymbol{\theta}^{(K)}$ (or the initial values for the first iteration),

$$Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(K)}) = E[\ell(\boldsymbol{y}_{obs}, \boldsymbol{y}_{mis}; \boldsymbol{\theta})\big|\boldsymbol{y}_{obs}; \boldsymbol{\theta}^{(K)}].$$

The M-step is then to find $\boldsymbol{\theta}^{(K+1)}$ to maximize $Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(K)})$ for all $\boldsymbol{\theta}$ in the parameter space,

$$Q(\boldsymbol{\theta}^{(K+1)}\big|\boldsymbol{\theta}^{(K)}) \geq Q(\boldsymbol{\theta}\big|\boldsymbol{\theta}^{(K)}).$$

The two steps are repeated until convergence is reached. When the complete data density belongs to the exponential family the E-step can be further simplified to compute the expected sufficient statistics of the complete data. After each iteration, the EM algorithm moves to a better point and the log-likelihood of the observed data increases.

For the multilevel nonlinear mixed effects model (4.2.1), assume a flat prior for the fixed effects $\boldsymbol{\beta}$ and consider $c(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)})$ as the complete data where $\boldsymbol{y}$ and $c(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)})$ are the observed and missing data, respectively. Let $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ represent the parameters for which REML estimates are required. The E-step and M-step of the FELA-EM algorithm are described as follows.

4.4.1 E-step

The E-step computes the conditional expectation of the complete data log-likelihood $\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta})$.

$$Q(\boldsymbol{\theta} \big| \boldsymbol{\theta}^{(K)}) = \int \ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}) p(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}) d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}, \tag{4.4.1}$$

where

$$\begin{aligned} \ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}) = & -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| \boldsymbol{y}_{ij} - f(\boldsymbol{\beta}, \boldsymbol{b}_i^{(1)}, \boldsymbol{b}_j^{(2)}) \right\|^2 \\ & - \frac{M_1}{2} \log(2\pi \left| \boldsymbol{D}^{(1)} \right|) - \frac{1}{2} \sum_{i=1}^{M_1} \boldsymbol{b}_i^{(1)^T} \boldsymbol{D}^{(1)^{-1}} \boldsymbol{b}_i^{(1)} \\ & - \frac{M_2}{2} \log(2\pi \left| \boldsymbol{D}^{(2)} \right|) - \frac{1}{2} \sum_{i=1}^{M_2} \boldsymbol{b}_j^{(2)^T} \boldsymbol{D}^{(2)^{-1}} \boldsymbol{b}_j^{(2)} \end{aligned} \tag{4.4.2}$$

and $N = \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} n_{ij}$ is the total number of observations. The density of the missing data $c(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)})$ conditional on the observed data $\boldsymbol{y}$ at $\boldsymbol{\theta}^{(K)}$, $p(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)})$, can be further written as

$$\begin{aligned} p(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}) &= \frac{p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \big| \boldsymbol{\theta}^{(K)})}{p(\boldsymbol{y} \big| \boldsymbol{\theta}^{(K)})} \\ &= \frac{\exp\{\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}^{(K)})\}}{\int \exp\{\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}^{(K)})\} d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}}. \end{aligned} \tag{4.4.3}$$

It can be seen from (4.4.2) that the density of the complete data $c(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)})$ is from the exponential family and the sufficient statistics of the complete data for $\boldsymbol{D}^{(1)}$, $\boldsymbol{D}^{(2)}$, and σ^2 are $S^{(1)}_{m_1,n_1} = \sum_{i=1}^{M_1} b^{(1)}_{i,m_1} b^{(1)}_{i,n_1} = \boldsymbol{b}^{(1)\prime}_{m_1} \boldsymbol{b}^{(1)}_{n_1}$ for $m_1, n_1 = 1, ..., q_1$, $S^{(2)}_{m_2,n_2} = \sum_{j=1}^{M_2} b^{(2)}_{j,m_2} b^{(2)}_{j,n_2} = \boldsymbol{b}^{(2)\prime}_{m_2} \boldsymbol{b}^{(2)}_{n_2}$ for $m_2, n_2 = 1, ..., q_2$, and $R = \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} \left\| \boldsymbol{y}_{ij} - f(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}_i, \boldsymbol{b}^{(2)}_j) \right\|^2 = \left\| \boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}) \right\|^2$, respectively. Thus, the E-step can be simplified to compute the expectations of R, $S^{(1)}_{m_1,n_1}$, and $S^{(2)}_{m_2,n_2}$ conditional on $\boldsymbol{y}$ at $\boldsymbol{\theta}^{(K)}$ that are given by

$$E(R \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}) = \frac{\int \left\| \boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}) \right\|^2 \exp\{\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}^{(K)})\} d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}}{\int \exp\{\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}^{(K)})\} d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}}, \tag{4.4.4}$$

$$E(S^{(1)}_{m_1,n_1}\left|\boldsymbol{y};\boldsymbol{\theta}^{(K)}\right.)=\frac{\int \boldsymbol{b}^{(1)\prime}_{m_1}\boldsymbol{b}^{(1)}_{n_1}\exp\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})\}d\boldsymbol{\beta}d\boldsymbol{b}^{(1)}d\boldsymbol{b}^{(2)}}{\int\exp\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})\}d\boldsymbol{\beta}d\boldsymbol{b}^{(1)}d\boldsymbol{b}^{(2)}},\tag{4.4.5}$$

and

$$E(S^{(2)}_{m_2,n_2}\left|\boldsymbol{y};\boldsymbol{\theta}^{(K)}\right.)=\frac{\int \boldsymbol{b}^{(2)\prime}_{m_2}\boldsymbol{b}^{(2)}_{n_2}\exp\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})\}d\boldsymbol{\beta}d\boldsymbol{b}^{(1)}d\boldsymbol{b}^{(2)}}{\int\exp\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})\}d\boldsymbol{\beta}d\boldsymbol{b}^{(1)}d\boldsymbol{b}^{(2)}},\tag{4.4.6}$$

where $m_1,n_1=1,...,q_1$, $m_2,n_2=1,...,q_2$, and

$$\begin{aligned}\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})=&-\frac{1}{2\{\sigma^2\}^{(K)}}\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\left\|\boldsymbol{y}_{ij}-f(\boldsymbol{\beta},\boldsymbol{b}_i^{(1)},\boldsymbol{b}_j^{(2)})\right\|^2\\&-\frac{1}{2}\sum_{i=1}^{M_1}\boldsymbol{b}_i^{(1)^T}\{\{\boldsymbol{D}^{(1)}\}^{(K)}\}^{-1}\boldsymbol{b}_i^{(1)}\\&-\frac{1}{2}\sum_{j=1}^{M_2}\boldsymbol{b}_j^{(2)^T}\{\{\boldsymbol{D}^{(2)}\}^{(K)}\}^{-1}\boldsymbol{b}_j^{(2)}.\end{aligned}\tag{4.4.7}$$

The common terms that appear in both the numerator and denominator in (4.4.4), (4.4.5), and (4.4.6) and do not contain $\boldsymbol{\beta}$, $\boldsymbol{b}^{(1)}$, and $\boldsymbol{b}^{(2)}$ are cancelled. The integrals in both the numerator and the denominator of (4.4.4), (4.4.5), and (4.4.6) generally cannot be computed analytically because both the sufficient statistics of the complete data for σ^2 (R) and $\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})$ can be nonlinear in the fixed and random effects. We

approximate (4.4.4), (4.4.5), and (4.4.6) by the fully exponential Laplace method that will be presented in Section 4.4.3.

4.4.2 M-step

The M-step is to find the estimates of the parameters for the next iteration, $\theta^{(K+1)}$, by solving the equation

$$\frac{\partial}{\partial \boldsymbol{\theta}} Q(\boldsymbol{\theta} \big| \boldsymbol{\theta}^{(K)}) = \frac{\partial}{\partial \boldsymbol{\theta}} \int \ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}; \boldsymbol{\theta}) p(\boldsymbol{\beta}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}) d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)} = 0. \tag{4.4.8}$$

By allowing differentiation under the integral sign (Walker 1996; Wang 2007), the unique solution to (4.4.8) is given by

$$\{\sigma^2\}^{(K+1)} = \frac{1}{N} E(R \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}), \tag{4.4.9}$$

$$\{D^{(1)}_{m_1,n_1}\}^{(K+1)} = \frac{1}{M_1} E(S^{(1)}_{m_1,n_1} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}), \qquad m_1, n_1 = 1,..., q_1, \tag{4.4.10}$$

and

$$\{D^{(2)}_{m_2,n_2}\}^{(K+1)} = \frac{1}{M_2} E(S^{(2)}_{m_2,n_2} \big| \boldsymbol{y}; \boldsymbol{\theta}^{(K)}), \qquad m_2, n_2 = 1,..., q_2. \tag{4.4.11}$$

Therefore, once the conditional expectations of the sufficient statistics $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$, $E(S^{(1)}_{m_1,n_1}|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ for $m_1,n_1=1,...,q_1$, and $E(S^{(2)}_{m_2,n_2}|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ for $m_2,n_2=1,...,q_2$ are obtained, the process of the M-step is straightforward.

4.4.3 Fully exponential Laplace approximation

The fully exponential Laplace approximation proposed by Tierney and Kadane (1986) is useful for approximating the expectation and variance of a strictly positive function. It has the advantage of requiring only second derivatives of the log-likelihood function to achieve a second-order accuracy. The fully exponential Laplace approximation to the ratio of two related integrals is given by

$$\begin{aligned}\frac{\int g(\boldsymbol{\varphi})\exp\{n\ell(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}}{\int \exp\{n\ell(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}} &= \frac{\int \exp\{n\ell^*(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}}{\int \exp\{n\ell(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}} \\ &\approx \left(\frac{\det\{-\partial^2\ell(\hat{\boldsymbol{\varphi}})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'\}}{\det\{-\partial^2\ell^*(\hat{\boldsymbol{\varphi}}^*)/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'\}}\right)^{1/2}\exp\{n\ell^*(\hat{\boldsymbol{\varphi}}^*)-n\ell(\hat{\boldsymbol{\varphi}})\},\end{aligned} \tag{4.4.12}$$

where $g(\boldsymbol{\varphi})$ is a positive scalar function, $\ell^*(\boldsymbol{\varphi}) = \log\{g(\boldsymbol{\varphi})\}/n + \ell(\boldsymbol{\varphi})$, and $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\varphi}}^*$ maximize ℓ and ℓ^*, respectively. Note that the errors in the Laplace approximations to both the numerator and the denominator in (4.12) are of order $O(1/n)$ while the error in the ratio (4.4.12) is of order $O(1/n^2)$ due to the cancellation of the similar error terms in the approximation to the numerator and the denominator. The limitation of the Laplace approximation in (4.4.12) is that it only applies to positive functions. Thus, it generally does not work for our problem because the covariance parameters can be negative and even the variance components are not necessarily strictly positive. They can take values close to zero or zero. Tierney, Kass and Kadane (1989) generalize the Laplace approximation in (4.4.12) so that $g(\boldsymbol{\varphi})$ can take on negative values. Their approach is to first approximate the moment generating function of $g(\boldsymbol{\varphi})$ (Note that the moment

generating function of any scalar function is always strictly positive), and then approximate the ratio in (4.4.12) by evaluating the first derivative of the approximation to the moment generating function at 0. They showed that the accuracy of the moment generating function approach is also of order $O(1/n^2)$.

We now apply the moment generating function approach to estimate the conditional expectations $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$, $E(S^{(1)}_{m_1,n_1}|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ for $m_1, n_1 = 1,...,q_1$, and $E(S^{(2)}_{m_2,n_2}|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ for $m_2, n_2 = 1,...,q_2$ given in (4.4.4), (4.4.5), and (4.4.6), respectively.

The moment generating function of $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ is given by

$$M_R(t) = \frac{\int \exp\left\{t\left\|\boldsymbol{y} - f(\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)})\right\|^2 + \hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})\right\} d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}}{\int \exp\left\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})\right\} d\boldsymbol{\beta} d\boldsymbol{b}^{(1)} d\boldsymbol{b}^{(2)}}, \quad (4.4.13)$$

with $\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)};\boldsymbol{\theta}^{(K)})$ as in (4.4.7). Let $\boldsymbol{\varphi} = c(\boldsymbol{\beta},\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)})$ and apply the fully exponential Laplace approximation (4.4.12) to (4.4.13) yielding

$$\begin{aligned}\widetilde{M}_R(t) = &\left(\frac{\det[-\partial^2 \hbar(\boldsymbol{y},\hat{\boldsymbol{\varphi}};\boldsymbol{\theta}^{(k)})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}{\det[-\partial^2\{t\left\|\boldsymbol{y} - f(\hat{\boldsymbol{\varphi}}^*)\right\|^2 + \hbar(\boldsymbol{y},\hat{\boldsymbol{\varphi}}^*;\boldsymbol{\theta}^{(k)})\}/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}\right)^{1/2} \\ &\times \exp\{t\left\|\boldsymbol{y} - f(\hat{\boldsymbol{\varphi}}^*)\right\|^2 + \hbar(\boldsymbol{y},\hat{\boldsymbol{\varphi}}^*;\boldsymbol{\theta}^{(k)}) - \hbar(\boldsymbol{y},\hat{\boldsymbol{\varphi}};\boldsymbol{\theta}^{(k)})\},\end{aligned} \quad (4.4.14)$$

where $\hat{\boldsymbol{\varphi}}^{*} = c(\hat{\boldsymbol{\beta}}^{*}, \hat{\boldsymbol{b}}^{(1)*}, \boldsymbol{b}^{(2)*})$ maximizes $t\left\|\boldsymbol{y} - f(\boldsymbol{\varphi})\right\|^2 + \hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$ and $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}^{(1)}, \boldsymbol{b}^{(2)})$ maximizes $\hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$. Then $E(R\big|\boldsymbol{y}; \boldsymbol{\theta}^{(k)})$ can be approximated by $\partial \widetilde{M}_R(t)/\partial t$ evaluated at 0. Since the analytic differentiation of $\widetilde{M}_R(t)$ with respect to t requires the third derivatives of $\left\|\boldsymbol{y} - f(\boldsymbol{\varphi})\right\|^2$ and $\hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$ with respect to $\boldsymbol{\varphi}$ that can be substantial work we apply the numerical differentiation approach to approximate $E(R\big|\boldsymbol{y}; \boldsymbol{\theta}^{(K)})$,

$$E(R\big|\boldsymbol{y}; \boldsymbol{\theta}^{(k)}) = \frac{\partial \widetilde{M}_R(0)}{\partial t} \approx \frac{\widetilde{M}_R(\delta) - \widetilde{M}_R(-\delta)}{2\delta},$$

for some small δ (i.e., 10^{-8}).

Similarly, one can show that the fully exponential Laplace approximations to the moment generating functions of $E(S^{(1)}_{m_1,n_1}\big|\boldsymbol{y}; \boldsymbol{\theta}^{(K)})$ for $m_1, n_1 = 1,..., q_1$ and $E(S^{(2)}_{m_2,n_2}\big|\boldsymbol{y}; \boldsymbol{\theta}^{(K)})$ for $m_2, n_2 = 1,..., q_2$ are given by

$$\widetilde{M}_{S^{(1)}_{m_1,n_1}}(t) = \left(\frac{\det[-\partial^2 \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}; \boldsymbol{\theta}^{(K)})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}{\det[-\partial^2 \{t\hat{\boldsymbol{b}}^{(1)**\prime}_{m_1} \hat{\boldsymbol{b}}^{(1)**}_{n_1} + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}^{**}; \boldsymbol{\theta}^{(K)})\}/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']} \right)^{1/2} \tag{4.4.15}$$

$$\times \exp\{t\hat{\boldsymbol{b}}^{(1)**\prime}_{m_1} \hat{\boldsymbol{b}}^{(1)**}_{n_1} + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}^{**}; \boldsymbol{\theta}^{(K)}) - \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}; \boldsymbol{\theta}^{(K)})\},$$

and

$$\widetilde{M}_{S^{(2)}_{m_2,n_2}}(t) = \left(\frac{\det[-\partial^2 \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}; \boldsymbol{\theta}^{(K)}) / \partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}']}{\det[-\partial^2 \{t\hat{\boldsymbol{b}}^{(2)***}_{m_2}{}' \hat{\boldsymbol{b}}^{(2)***}_{n_2} + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}^{***}; \boldsymbol{\theta}^{(K)})\} / \partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}']} \right)^{1/2}$$

$$\times \exp\{t\hat{\boldsymbol{b}}^{(2)***}_{m_2}{}' \hat{\boldsymbol{b}}^{(2)***}_{n_2} + \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}^{***}; \boldsymbol{\theta}^{(K)}) - \hbar(\boldsymbol{y}, \hat{\boldsymbol{\varphi}}; \boldsymbol{\theta}^{(K)})\}, \tag{4.4.16}$$

where $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}^{(1)}, \hat{\boldsymbol{b}}^{(2)})$ maximizes $\hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$, $\hat{\boldsymbol{\varphi}}^{**} = c(\hat{\boldsymbol{\beta}}^{**}, \hat{\boldsymbol{b}}^{(1)**}, \hat{\boldsymbol{b}}^{(2)**})$ maximizes $t\boldsymbol{b}^{(1)}_{m_1}{}' \boldsymbol{b}^{(1)}_{n_1} + \hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$, and $\hat{\boldsymbol{\varphi}}^{***} = c(\hat{\boldsymbol{\beta}}^{***}, \hat{\boldsymbol{b}}^{(2)***}, \hat{\boldsymbol{b}}^{(2)***})$ maximizes $t\boldsymbol{b}^{(2)}_{m_2}{}' \boldsymbol{b}^{(2)}_{n_2} + \hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$. As noted by Tierney and Kadane (1986), the computation of the fully exponential Laplace approximation is relatively minimal since 1) once $\hat{\boldsymbol{\varphi}}$, the maximum of $\hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$, has been determined, it can be used as starting values to find $\hat{\boldsymbol{\varphi}}^*$, $\hat{\boldsymbol{\varphi}}^{**}$, and $\hat{\boldsymbol{\varphi}}^{***}$, the maximum of $t\|\boldsymbol{y} - f(\boldsymbol{\varphi})\|^2 + \hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$, $t\boldsymbol{b}^{(1)}_{m_1}{}' \boldsymbol{b}^{(1)}_{n_1} + \hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$, and $t\boldsymbol{b}^{(2)}_{m_2}{}' \boldsymbol{b}^{(2)}_{n_2} + \hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$ in (4.4.14), (4.4.15), and (4.4.16), respectively; 2) generally, the number of iterations needed to find $\hat{\boldsymbol{\varphi}}^*$, $\hat{\boldsymbol{\varphi}}^{**}$, and $\hat{\boldsymbol{\varphi}}^{***}$ from $\hat{\boldsymbol{\varphi}}$ is quite small and replacing $\hat{\boldsymbol{\varphi}}^*$, $\hat{\boldsymbol{\varphi}}^{**}$, and $\hat{\boldsymbol{\varphi}}^{***}$ by two Newton steps from $\hat{\boldsymbol{\varphi}}$ are usually sufficient.

In summary, the FELA-EM algorithm for multilevel nonlinear mixed models with two crossed groupings takes the following steps:

1) Initialize $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)}) = \boldsymbol{\theta_0}$.

2) E-step:

   a) Find $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}}^{(1)}, \hat{\boldsymbol{b}}^{(2)})$ to maximize $\hbar(\boldsymbol{y}, \boldsymbol{\varphi}; \boldsymbol{\theta}^{(K)})$

b) Maximize $t\|\boldsymbol{y}-f(\boldsymbol{\varphi})\|^2+\hbar(\boldsymbol{y},\boldsymbol{\varphi};\boldsymbol{\theta}^{(K)})$, $t\boldsymbol{b}_{m_1}^{(1)'}\boldsymbol{b}_{n_1}^{(1)}+\hbar(\boldsymbol{y},\boldsymbol{\varphi};\boldsymbol{\theta}^{(K)})$, and $t\boldsymbol{b}_{m_2}^{(2)'}\boldsymbol{b}_{n_2}^{(2)}+\hbar(\boldsymbol{y},\boldsymbol{\varphi};\boldsymbol{\theta}^{(K)})$ by replacing $\hat{\boldsymbol{\varphi}}^{*}=c(\hat{\boldsymbol{\beta}}^{*},\hat{\boldsymbol{b}}^{(1)*},\hat{\boldsymbol{b}}^{(2)*})$, $\hat{\boldsymbol{\varphi}}^{**}=c(\hat{\boldsymbol{\beta}}^{**},\hat{\boldsymbol{b}}^{(1)**},\hat{\boldsymbol{b}}^{(2)**})$, and $\hat{\boldsymbol{\varphi}}^{***}=c(\hat{\boldsymbol{\beta}}^{***},\hat{\boldsymbol{b}}^{(1)***},\hat{\boldsymbol{b}}^{(2)***})$ with two Newton steps from $\hat{\boldsymbol{\varphi}}$, respectively.

c) Approximate the conditional expectations $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$, $E(S_{m_1,n_1}^{(1)}|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ for $m_1,n_1=1,...,q_1,$ and $E(S_{m_2,n_2}^{(2)}|\boldsymbol{y};\boldsymbol{\theta}^{(K)})$ for $m_2,n_2=1,...,q_2$ with the numerical differentiation approach.

3) M-step: Obtain $\boldsymbol{\theta}^{(K+1)}=c\left(\{\sigma^2\}^{(K+1)},\{D^{(1)}\}^{(K+1)},\{D^{(2)}\}^{(K+1)}\right)$ according to (4.4.9), (4.4.10), and (4.4.11), respectively.

4) Repeat steps 2 and 3 until convergence.

We conclude convergence when either the difference between two successive log-likelihood of model (4.2.1) or the absolute/relative changes in $\boldsymbol{\theta}$ are less than 10^{-6}. The convergence rate of the standard EM algorithm is slow (McLachlan and Krishnan, 2008). When the absolute/relative changes in $\boldsymbol{\theta}$ are less than 10^{-3}, we switch to the accelerated version of the EM algorithm, algorithm QN1, proposed by Jamshidian and Jennrich (1997). The QN1 algorithm is based on a quasi-Newton method, Broyden's method, for solving nonlinear equations and minimizing functions. Algorithm QN1 is easy to implement and its speed of convergence can be 19-87 times faster than that of the standard EM algorithm.

4.4.4 Calculating the information matrix

One of the drawbacks of EM algorithm is that it does not provide a natural estimator for the information matrix and thus the standard errors of the estimates cannot be readily obtained. Let $\hat{\boldsymbol{\theta}}$ be the estimates of $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ at convergence. The observed information matrix of $\hat{\boldsymbol{\theta}}$ can be obtained by directly maximizing the logarithm of the Laplace approximated REML version of observed data likelihood $p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ given in (4.3.3) with respect to $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ by one quasi-Newton step from $\hat{\boldsymbol{\theta}}$.

4.4.5 Estimating the fixed and random effects

The proposed FELA-EM algorithm is a REML method for estimating variance and covariance parameters $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ in the nonlinear mixed model (4.2.1) and it does not provide estimates for both the fixed effects $\boldsymbol{\beta}$ and the random effects $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$. We can directly maximize the logarithm of the Laplace approximated ML version of observed data likelihood $p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ given in (4.3.2) with respect to both $\boldsymbol{\beta}$, $\boldsymbol{b}^{(1)}$, and $\boldsymbol{b}^{(2)}$ while holding $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)})$ at $\hat{\boldsymbol{\theta}}$, the estimates at convergence from the FELA-EM algorithm, to obtain the "REML estimates" of $\boldsymbol{\beta}$, $\boldsymbol{b}^{(1)}$, and $\boldsymbol{b}^{(2)}$ in model (4.2.1).

4.5 Comparing the approximations

In this section, we present a comparison of our proposed EM algorithm (FELA-EM-REML), the ML version Laplace approximation (Laplace-ML), and four variations of linearization methods described in Section 4.1 via simulation studies. We restrict ourselves to the linearization methods proposed by Wolfinger and Lin (1997). Both ML and REML estimates are obtained for the linearization methods, denoted as ZERO-ML, ZERO-REML, EBLUP-ML and EBLUP-REML, respectively. Two models are used in the simulation studies, a logistic model and a first-order compartment model. Both models are widely used in statistical literatures to illustrate the fitting of nonlinear mixed models (Lindstrom and Bates, 1990; Pinheiro and Bates, 1995; Wolfinger and Lin, 1997; Kuhn and Lavielle, 2005; Wang, 2007). For both models 2,000 simulated data sets from two random number seeds (1,000 data sets/seed) are generated to avoid simulation bias and ML/REML estimates using different approximations are obtained. The FELA-EM algorithm presented here is done using the code written in R by the authors. The Laplace approximation is done using the nlmer function in R package lme4 (Bates, Maechler, and Dai, 2008) and the four linearization methods (ZERO-ML, ZERO-REML, EBLUP-ML and EBLUP-REML) are done using the %nlinmix macro in SAS. The sample code for fitting the logistic model using the FELA-EM algorithm is given in the appendix.

4.5.1 Logistic model

A 3-parameter logistic model used by Wolfinger and Lin (1997), but with two crossed grouping factors and two random effects associated with each of the grouping factor was used to generate the data. The values of fixed-effects parameters, covariate, and

variance-covariance parameters for random effects were also similar to those used by Wolfinger and Lin (1997). The error σ^2 was increased from 25, used in Wolfinger and Lin (1997), to 625 to distinguish the performance of different approximation procedures. The nonlinear mixed model is given by

$$y_{ijk} = \frac{\beta_1 + b_{i1}^{(1)} + b_{j1}^{(2)}}{1 + \exp\{-[x_{ij} - (\beta_2 + b_{i2}^{(1)} + b_{j2}^{(2)})]/\beta_3\}} + \varepsilon_{ijk}, \qquad (4.4.1)$$
$$i = 1,...,M_1, \quad j = 1,...,M_2, \quad k = 1,...,n_{ij},$$

where $\boldsymbol{b}_i^{(1)} = (b_{i1}^{(1)}, b_{i2}^{(1)})^T$ are i.i.d. $N(0, \boldsymbol{D}^{(1)})$, $\boldsymbol{b}_j^{(2)} = (b_{j1}^{(2)}, b_{j2})^{(2)T}$ are i.i.d. $N(0, \boldsymbol{D}^{(2)})$ and independent of the $\boldsymbol{b}_i^{(1)}$, and ε_{ijk} are i.i.d. $N(0, \sigma^2)$ and independent of $\boldsymbol{b}_i^{(1)}$ and $\boldsymbol{b}_j^{(2)}$. We use $M_1 = 10$, $M_2 = 10$, $n_{ij} = 7$ for $i = 1,...,M_1$ and $j = 1,...,M_2$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^T = (200, 700, 350)^T$, and $\boldsymbol{x}_{ij} = (118, 484, 664, 1004, 1231, 1372, 1582)^T$. For variance and covariance parameters, we let

$$\boldsymbol{D}^{(1)} = \begin{bmatrix} D_{11}^{(1)} & D_{12}^{(1)} \\ D_{21}^{(1)} & D_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} 2500 & -1875 \\ -1875 & 15625 \end{bmatrix},$$

and

$$\boldsymbol{D}^{(2)} = \begin{bmatrix} D_{11}^{(2)} & D_{12}^{(2)} \\ D_{21}^{(2)} & D_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} 2600 & -1950 \\ -1950 & 16250 \end{bmatrix}.$$

Tables 4.1 lists the simulation results for the fixed-effects estimates in the logistic model. Assuming $\hat{\theta}_r$ stands for a parameter estimate at the r^{th} simulation and θ_T for the true value of the parameter, the summary statistics for the parameters are defined as follows: Mean denotes the average of the estimates $\hat{\theta}_r$ across the 2000 simulations, i.e., $\sum_{r=1}^{2000} \hat{\theta}_r / 2000$; %Bias denotes the relative bias in estimating the parameter, i.e., $100 \times (\text{Mean} - \theta_T) / |\theta_T|$; RMSE denotes the square root of mean square error of the estimator, i.e., $\sqrt{\sum_{r=1}^{2000} (\hat{\theta}_r - \theta_T)^2 / 2000}$, which is a measure of accuracy that takes into account both bias and variability; 95% CI denotes 95% confidence intervals, i.e., $\text{Mean} \pm \text{Z}_\alpha \sqrt{s_{\hat{\theta}}^2 / 2000}$, where Z_α is the normal critical value corresponding to 95% confidence level and $s_{\hat{\theta}}^2 = \sum_{r=1}^{2000} (\hat{\theta}_r - \theta_T)^2 / (2000 - 1)$. When 95% CI does not cover the true value of the parameter, we conclude that the estimate is significantly biased at 5% level; and finally, %CVR denotes the observed coverage of the standard normal based 95% confidence intervals computed using the model-based standard errors. Only those values for the fixed effects were presented in the paper because the standard errors for the variance-covariance parameters were not provided by both the nlmer function in R and the %nlinmix macro in SAS. The 95% coverage values marked with an asterisk are outside the interval (93.54, 96.46). The half-width of this interval is three times the binomial standard error, which is $[(95)(5)/2000]^{1/2} = 0.4873$. For β_1, which enters the nonlinear mixed model linearly and is associated with the random effects, the estimates obtained from FELA-EM-REML, Laplace-ML, and the two zero-expansion approximations (ZERO-ML and ZERO-REML) are more accurate than those from the

two eblup-expansion approximations (EBLUP-ML and EBLUP-REML). The two eblup-expansion approximations significantly underestimate the estimator by 1.61% and 1.54%, respectively, while the other four approximations all provide unbiased estimates. For β_2, which enters the model nonlinearly and is associated with the random effects, the estimates obtained from FELA-EM-REML and Laplace-ML approximations are more accurate than those from the four linearization methods. Although the 95% confidence intervals show that all six approximations produce significantly biased estimates, the %Bias are relatively small with an absolute maximum value 2.22% for EBLUP-ML approximation. For β_3, which enters the model nonlinearly and is the only non-random coefficient, the estimates obtained from FELA-EM-REML and Laplace-ML approximations are again more accurate than those from the four linearization methods. The two zero-expansion approximations (ZERO-ML and ZERO-REML) significantly overestimate β_3 by 5.29% while the two eblup-expansion methods (EBLUP-ML and EBLUP-REML) underestimate the parameter by 3.94% and 3.81%, respectively. Although FELA-EM-REML and Laplace-ML approximations give significantly positive biased estimates for β_3, the %Bias is 0.89% and 0.67%, respectively, which is more than three times smaller than those from the four linearization methods. The two approximations that give unbiased or close-to-unbiased (%Bias < 1) estimates for all three fixed effects are FELA-EM-REML and Laplace-ML approximations. The observed 95% confidence interval coverages obtained from FELA-EM-REML approximation are more accurate than those from the other five methods (ZERO-ML, EBLUP-ML, Laplace-ML, ZERO-REML, and EBLUP-REML). For β_1, the EBLUP-ML method gives a considerable lower coverage rate while the other five approximations all attain their

nominal coverages. For β_2, the only approximation method which gives similar coverage to the nominal value is FELA-EM-REML while the observed coverages tend to decrease from the nominal one for the other five methods. For β_3, both Laplace-ML and FELA-EM-REML approximations attain their nominal coverages while the other four approximations all provide a significantly lower coverage rate compared to the nominal value. The square root of mean square errors (RMSE) of β_1 and β_2 are relatively similar for all six approximations considered. For β_3, the RMSE's for the two zero-expansion approximations (ZERO-ML and ZERO-RML) are roughly 30% larger than those from the other four approximations.

Table 4.1: Simulation results for the fixed effects in the logistic model

Approximation	Mean	%Bias	RMSE	95% CI	%CVR
$\beta_1 = 200$					
ZERO-ML	200.77	0.38	23.18	[199.75, 201.78]	94.21
EBLUP-ML	196.77†	-1.61	23.37	[195.76, 197.79]	93.12*
Laplace-ML	200.24	0.12	22.99	[199.23, 201.25]	93.90
ZERO-REML	200.77	0.38	23.18	[199.75, 201.78]	94.62
EBLUP-REML	196.92†	-1.54	23.35	[195.90, 197.93]	94.08
FELA-EM-REML	200.45	0.22	22.99	[199.44, 201.46]	94.85
$\beta_2 = 700$					
ZERO-ML	685.31†	-2.10	67.07	[682.44, 688.18]	91.84*
EBLUP-ML	684.46†	-2.22	61.58	[681.85, 687.07]	92.35*
Laplace-ML	702.81†	0.40	64.04	[700.01, 705.61]	92.80*
ZERO-REML	685.31†	-2.10	67.07	[682.44, 688.18]	92.60*
EBLUP-REML	685.02†	-2.14	61.57	[682.41, 687.64]	92.91*
FELA-EM-REML	703.63†	0.52	64.35	[700.81, 706.44]	93.55
$\beta_3 = 350$					
ZERO-ML	368.52†	5.29	28.53	[367.57, 369.47]	89.77*
EBLUP-ML	336.22†	-3.94	21.67	[335.48, 336.95]	86.29*
Laplace-ML	352.36†	0.67	18.38	[351.56, 353.16]	95.70
ZERO-REML	368.52†	5.29	28.53	[367.57, 369.47]	89.83*
EBLUP-REML	336.68†	-3.81	21.32	[335.95, 337.41]	86.99*
FELA-EM-REML	353.12†	0.89	18.62	[352.32, 353.93]	95.65

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

* The %CVR is outside the interval (93.54, 96.46).

Table 4.2 contains the simulation results for the variance-covariance components in the logistic model. For variance components associated with the random effects that enter the nonlinear mixed model linearly ($D_{11}^{(1)}$ and $D_{11}^{(2)}$), the estimate obtained from FELA-EM-REML approximation has the smallest bias for both $D_{11}^{(1)}$ and $D_{11}^{(2)}$. All three ML approximations (ZERO-ML, EBLUP-ML, and Laplace-ML) significantly underestimate both parameters ($D_{11}^{(1)}$ and $D_{11}^{(2)}$) with a minimum %Bias 4.77% for ZERO-ML when estimating $D_{11}^{(1)}$; while the three REML approximations (ZERO-REML, EBLUP-REML, and FELA-EM-REML) all give unbiased estimates with a maximum absolute %Bias 1.68% for EBLUP-REML for estimating $D_{11}^{(2)}$. For variance components associated with the random effects that enter the nonlinear mixed model nonlinearly ($D_{22}^{(1)}$ and $D_{22}^{(2)}$), the only approximation that gives unbiased estimates for both $D_{22}^{(1)}$ and $D_{22}^{(2)}$ is FELA-EM-REML with a absolute %Bias of 0.38 and 0.69 for estimating $D_{22}^{(1)}$ and $D_{22}^{(2)}$, respectively. All three ML approximations (ZERO-ML, EBLUP-ML, and Laplace-ML) again give significantly negative biased estimates with a maximum absolute %Bias 13.62% for EBLUP-ML when estimating $D_{22}^{(1)}$; while the results for the two linearization REML approximations (ZERO-REML and EBLUP-REML) are different. The ZERO-REML approximation significantly overestimates $D_{22}^{(1)}$ and $D_{22}^{(2)}$ by 3.83% and 4.61%, respectively while the EBLUP-REML approximation significantly underestimates $D_{22}^{(1)}$ and $D_{22}^{(2)}$ by 6.48% and 5.54%, respectively. The results for covariance components ($D_{12}^{(1)}$ and $D_{12}^{(2)}$) are different from each other. For $D_{12}^{(1)}$, the EBLUP-ML approximation and the two ZERO approximations (ZERO-ML and ZERO-REML) significantly overestimate the parameter by 6.17%, 15.48%, and 11.68%, respectively while the other

three approximations (Laplace-ML, EBLUP-REML, and FELA-EM-REML) all give unbiased estimates with a maximum absolute %Bias of 3.60% for FELA-EM-REML. For $D_{12}^{(2)}$, FELA-EM-REML is the only approximation that gives an unbiased estimate while the other five approximations all significantly overestimate the parameter with a minimum %Bias of 5.34% for Laplace-ML. For the subject specific variance (σ^2) estimation, the estimate obtained from the FELA-EM-REML approximation has the smallest absolute %Bias. Both ZERO approximations (ZERO-ML and ZERO-REML) significantly overestimate the estimators by 3.33% and 3.47%, respectively while the other four approximations all give unbiased estimates. The RMSE's of the variance-covariance components for the random effects are relatively similar for all six approximations considered. For the subject specific variance, σ^2, the RMSE's for the two zero-expansion approximations (ZERO-ML and ZERO-RML) can be 20% larger than those from the other four approximations.

Table 4.2: Simulation results for the variance-covariance components in the logistic model

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11}^{(1)} = 2500$				
ZERO-ML	2380.77†	-4.77	1156.50	[2330.34, 2431.20]
EBLUP-ML	2320.64†	-7.17	1130.36	[2271.72, 2369.56]
Laplace-ML	2365.50†	-5.38	1145.93	[2315.61, 2415.39]
ZERO-REML	2536.82	1.47	1250.57	[2482.03, 2591.62]
EBLUP-REML	2474.28	-1.03	1214.53	[2421.05, 2527.51]
FELA-EM-REML	2513.06	0.52	1236.47	[2458.86, 2567.26]
$D_{12}^{(1)} = -1875$				
ZERO-ML	-1584.84†	15.48	2168.42	[-1679.04, -1490.64]
EBLUP-ML	-1759.22†	6.17	2100.43	[-1851.16, -1667.28]
Laplace-ML	-1848.94	1.39	2194.75	[-1945.14, -1752.73]
ZERO-REML	-1655.99†	11.68	2354.78	[-1758.77, -1553.21]
EBLUP-REML	-1849.64	1.35	2293.72	[-1950.19, -1749.10]
FELA-EM-REML	-1942.56	-3.60	2395.32	[-2047.52, -1837.60]
$D_{22}^{(1)} = 15625$				
ZERO-ML	14995.95†	-4.03	9626.84	[14574.83, 15417.06]
EBLUP-ML	13496.89†	-13.62	7797.83	[13168.03, 13825.75]
Laplace-ML	14535.80†	-6.97	8125.87	[14182.80, 14888.80]
ZERO-REML	16223.19†	3.83	10506.15	[15763.38, 16683.01]
EBLUP-REML	14611.80†	-6.48	8261.24	[14252.39, 14971.22]
FELA-EM-REML	15565.90	-0.38	8794.90	[15180.37, 15951.44]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

Table 4.2 (continued): Simulation results for the variance-covariance components in the logistic model

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11}^{(2)} = 2600$				
ZERO-ML	2457.13†	-5.49	1179.50	[2405.81, 2508.46]
EBLUP-ML	2394.28†	-7.91	1153.53	[2344.52, 2444.03]
Laplace-ML	2441.36†	-6.10	1169.81	[2390.55, 2492.17]
ZERO-REML	2621.89	0.84	1274.79	[2566.01, 2677.76]
EBLUP-REML	2556.44	-1.68	1237.81	[2502.22, 2610.67]
FELA-EM-REML	2597.47	-0.10	1261.94	[2542.15, 2652.79]
$D_{12}^{(2)} = -1950$				
ZERO-ML	-1562.23†	19.89	2185.52	[-1656.52, -1467.95]
EBLUP-ML	-1755.07†	10.00	2129.31	[-1848.02, -1662.12]
Laplace-ML	-1845.87†	5.34	2220.77	[-1943.12, -1748.63]
ZERO-REML	-1630.01†	16.41	2371.62	[-1733.02, -1526.99]
EBLUP-REML	-1840.48†	5.62	2318.91	[-1942.02, -1738.94]
FELA-EM-REML	-1935.95	0.72	2423.62	[-2042.19, -1829.71]
$D_{22}^{(2)} = 16250$				
ZERO-ML	15687.81†	-3.46	10004.15	[15249.95, 16125.67]
EBLUP-ML	14153.24†	-12.90	8080.46	[13811.15, 14495.34]
Laplace-ML	15245.90†	-6.18	8479.55	[14876.80, 15615.01]
ZERO-REML	16998.83†	4.61	10947.57	[16520.05, 17477.62]
EBLUP-REML	15350.00†	-5.54	8604.18	[14974.88, 15725.11]
FELA-EM-REML	16362.10	0.69	9215.58	[15958.15, 16766.06]
$\sigma^2 = 625$				
ZERO-ML	645.82†	3.33	42.14	[644.22, 647.43]
EBLUP-ML	625.19	0.03	34.20	[623.69, 626.69]
Laplace-ML	624.65	-0.06	34.08	[623.16, 626.15]
ZERO-REML	646.66†	3.47	42.60	[645.05, 648.27]
EBLUP-REML	625.93	0.15	34.23	[624.43, 627.43]
FELA-EM-REML	624.95	-0.01	34.09	[623.46, 626.44]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

4.5.2 First-order compartment model

A 3-parameter first-order compartment model used by Pinheiro and Bates (1995), but with two crossed grouping factors and two random effects associated with each of the grouping factor was used to generate the data. The values of the fixed-effects parameters, the variance-covariance parameters, the covariate, and the dose were also similar to those used by Pinheiro and Bates (1995). The nonlinear mixed model is given by

$$y_{ijk} = \frac{\text{Dose} \cdot \exp(-(\beta_1 + b_{i1}^{(1)} + b_{j1}^{(2)}) + (\beta_2 + b_{i2}^{(1)} + b_{j2}^{(2)}) + \beta_3)}{\exp(\beta_2 + b_{i2}^{(1)} + b_{j2}^{(2)}) - \exp(\beta_3)} \{\exp[-\exp(\beta_3)x_{ijk}]$$

$$- \exp[-\exp(\beta_2 + b_{i2}^{(1)} + b_{j2}^{(2)})x_{ijk}]\} + \varepsilon_{ijk},$$

$$i = 1,...,M_1, \quad j = 1,...,M_2, \quad k = 1,...,n_{ij},$$

(4.5.2)

where $\boldsymbol{b}_i^{(1)} = (b_{i1}^{(1)}, b_{i2}^{(1)})^T$ are i.i.d. $N(0, \boldsymbol{D}^{(1)})$, $\boldsymbol{b}_j^{(2)} = (b_{j1}^{(2)}, b_{j2}^{(2)})^T$ are i.i.d. $N(0, \boldsymbol{D}^{(2)})$ and independent of the $\boldsymbol{b}_i^{(1)}$, and ε_{ijk} are i.i.d. $N(0, \sigma^2)$ and independent of $\boldsymbol{b}_i^{(1)}$ and $\boldsymbol{b}_j^{(2)}$. We use Dose = 1, $M_1 = 12$, $M_2 = 12$, $n_{ij} = 11$ for $i = 1,...,M_1$ and $j = 1,...,M_2$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^T = (-3, 0.5, -2.5)^T$, $\boldsymbol{x}_{ij} = (0, 0.25, 0.5, 1, 2, 4, 5, 7, 9, 12, 24)^T$, $\sigma^2 = 0.25$,

$$\boldsymbol{D}^{(1)} = \begin{bmatrix} D_{11}^{(1)} & D_{12}^{(1)} \\ D_{21}^{(1)} & D_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.25 \end{bmatrix}, \text{ and } \boldsymbol{D}^{(2)} = \begin{bmatrix} D_{11}^{(2)} & D_{12}^{(2)} \\ D_{21}^{(2)} & D_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} 0.06 & 0 \\ 0 & 0.3 \end{bmatrix}.$$

Table 4.3 summarizes the simulation results for the fixed-effects estimates. For β_1, the two zero-expansion approximations significantly underestimate the parameter by 3.34%

while the other four approximations show very little negative bias with a maximum absolute value of %Bias 0.32% for Laplace-ML. For β_2, the two zero-expansion approximations highly overestimate the parameter by more than 18% while the two eblup-expansion approximations significantly underestimate the estimator by 6.82% and 6.79%, respectively. The two approximations that give unbiased estimates for β_2 are FELA-EM-REML and Laplace-ML. For the non-random coefficient β_3, the results are similar to those for β_1. Both zero-expansion approximations significantly underestimate the estimator with %Bias by 4.24% while the other four approximations provide a small but significant bias with a maximum absolute value of %Bias 0.52% for EBLUP-REML. For the observed 95% confidence interval coverages, the only approximation method which gives similar coverages to the nominal values for all three fixed effects is FELA-EM-REML. For β_1, all six approximations considered attain their nominal coverages. For β_2, both EBLUP-REML and FELA-EM-REML provide values similar to the nominal coverages while the other four approximations (ZERO-ML, EBLUP-ML, Laplace-ML, and ZERO-REML) all give significantly lower coverage rates. For β_3, the observed coverages obtained from ZERO-REML and the two ML linearization methods (ZERO-ML and EBLUP-ML) are significantly lower than the nominal values while the other three methods (Laplace-ML, EBLUP-REML, and FELA-EM-REML) all give similar coverages to their nominal ones. We also note that the RMSE of both β_1 and β_2 is relatively similar for all approximations considered and the RMSE of β_3 for the two zero-expansion approximations is about twice as large as that for the other four approximations.

Table 4.3: Simulation results for the fixed effects in the first-order compartment model

Approximation	Mean	%Bias	RMSE	95% CI	%CVR
$\beta_1 = -3$					
ZERO-ML	-3.1003†	-3.34	0.1473	[-3.1051, -3.0956]	94.41
EBLUP-ML	-3.0048†	-0.16	0.1047	[-3.0094, -3.0002]	93.68
Laplace-ML	-3.0095†	-0.32	0.1048	[-3.0141, -3.0050]	94.60
ZERO-REML	-3.1002†	-3.34	0.1472	[-3.1049, -3.0954]	95.12
EBLUP-REML	-3.0042	-0.14	0.1047	[-3.0088, -2.9996]	94.75
FELA-EM-REML	-3.0091†	-0.30	0.1048	[-3.0136, -3.0045]	95.10
$\beta_2 = 0.5$					
ZERO-ML	0.5902†	18.04	0.2304	[0.5808, 0.5995]	92.14*
EBLUP-ML	0.4659†	-6.82	0.2277	[0.4560, 0.4759]	92.46*
Laplace-ML	0.4984	-0.31	0.2288	[0.4884, 0.5085]	93.30*
ZERO-REML	0.5903†	18.06	0.2302	[0.5810, 0.5996]	93.00*
EBLUP-REML	0.4661†	-6.79	0.2280	[0.4561, 0.4760]	94.08
FELA-EM-REML	0.5008	0.17	0.2295	[0.4908, 0.5109]	94.15
$\beta_3 = -2.5$					
ZERO-ML	-2.6059†	-4.24	0.1268	[-2.6090, -2.6028]	90.58*
EBLUP-ML	-2.4872†	0.51	0.0590	[-2.4897, -2.4847]	86.85*
Laplace-ML	-2.5114†	-0.46	0.0588	[-2.5139, -2.5089]	95.95
ZERO-REML	-2.6060†	-4.24	0.1269	[-2.6090, -2.6029]	90.64*
EBLUP-REML	-2.4871†	0.52	0.0589	[-2.4896, -2.4845]	96.28
FELA-EM-REML	-2.5116†	-0.46	0.0588	[-2.5141, -2.5090]	96.25

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

* The %CVR is outside the interval (93.54, 96.46).

Table 4.4 shows the simulation results for the variance-covariance components in the first-order compartment model. For variance components associated with the random effects $\boldsymbol{b}^{(1)}$ ($D_{11}^{(1)}$ and $D_{11}^{(2)}$), the three ML approximations (ZERO-ML, EBLUP-ML, and Laplace-ML) significantly underestimate the estimators with a minimum %Bias 2.93% for ZERO-ML when estimating $D_{11}^{(2)}$ while FELA-EM-REML and EBLUP-REML approximations both give unbiased estimates with a maximum absolute %Bias 1.68% for EBLUP-REML for estimating $D_{11}^{(1)}$. The results for ZERO-REML are different when estimating $D_{11}^{(1)}$ and $D_{11}^{(2)}$. It produces an unbiased estimate for $D_{11}^{(1)}$ while significantly overestimates $D_{11}^{(2)}$ by 2.68%. For variance components associated with the random effects $\boldsymbol{b}^{(2)}$ ($D_{22}^{(1)}$ and $D_{22}^{(2)}$), the only approximation that gives unbiased estimates is FELA-EM-REML while the other five approximations all give significantly negative biased estimates. The two zero-expansion approximations (ZERO-ML and ZERO-REML) highly underestimate the parameter by more than 20% and the two eblup-expansion approximations and Laplace-ML all show a mild bias with a minimum absolute %Bias of 3.13% for EBLUP-REML when estimating $D_{22}^{(1)}$. For covariance components $D_{12}^{(1)}$ and $D_{12}^{(2)}$, the two zero-expansion approximations (ZERO-ML and ZERO-REML) significantly underestimate the estimators while the other approximations all give unbiased estimates. For the subject specific variance (σ^2) estimation, the two zero-expansion approximations (ZERO-ML and ZERO-REML) significantly overestimate the estimator by 3.2% and 3.26%, respectively while the other approximations again all produce unbiased estimates with a maximum absolute %Bias

0.1% for Laplace-ML. The RMSE of the variance-covariance components is relatively similar for all approximations considered.

Table 4.4: Simulation results for the variance-covariance components in the first-order compartment model

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11}^{(1)} = 0.05$				
ZERO-ML	0.0484†	-3.22	0.0235	[0.0474, 0.0494]
EBLUP-ML	0.0467†	-6.55	0.0215	[0.0458, 0.0477]
Laplace-ML	0.0475†	-4.97	0.0218	[0.0466, 0.0485]
ZERO-REML	0.0509	1.73	0.0251	[0.0498, 0.0520]
EBLUP-REML	0.0492	-1.68	0.0228	[0.0482, 0.0502]
FELA-EM-REML	0.0500	0.00	0.0232	[0.0490, 0.0510]
$D_{12}^{(1)} = 0$				
ZERO-ML	-0.0061†	NA	0.0336	[-0.0076, -0.0046]
EBLUP-ML	-0.0002	NA	0.0335	[-0.0017, 0.0013]
Laplace-ML	-0.0005	NA	0.0344	[-0.0020, 0.0010]
ZERO-REML	-0.0060†	NA	0.0359	[-0.0076, -0.0045]
EBLUP-REML	0.0002	NA	0.0360	[-0.0014, 0.0018]
FELA-EM-REML	0.0005	NA	0.0372	[-0.0011, 0.0022]
$D_{22}^{(1)} = 0.25$				
ZERO-ML	0.1832†	-26.73	0.1133	[0.1791, 0.1872]
EBLUP-ML	0.2284†	-8.64	0.1117	[0.2236, 0.2332]
Laplace-ML	0.2321†	-7.17	0.1153	[0.2271, 0.2371]
ZERO-REML	0.1943†	-22.30	0.1136	[0.1899, 0.1986]
EBLUP-REML	0.2422†	-3.13	0.1177	[0.2370, 0.2473]
FELA-EM-REML	0.2487	-0.50	0.1240	[0.2433, 0.2542]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

Table 4.4 (continued): Simulation results for the variance-covariance components in the first-order compartment model

Approximation	Mean	%Bias	RMSE	95% CI
$D_{11}^{(2)} = 0.06$				
ZERO-ML	0.0582†	-2.93	0.0270	[0.0571, 0.0594]
EBLUP-ML	0.0562†	-6.41	0.0243	[0.0551, 0.0572]
Laplace-ML	0.0570†	-5.06	0.0246	[0.0559, 0.0580]
ZERO-REML	0.0616†	2.68	0.0290	[0.0603, 0.0629]
EBLUP-REML	0.0594	-1.00	0.0258	[0.0583, 0.0605]
FELA-EM-REML	0.0603	0.57	0.0263	[0.0592, 0.0615]
$D_{12}^{(2)} = 0$				
ZERO-ML	-0.0063†	NA	0.0413	[-0.0081, -0.0045]
EBLUP-ML	0.0000	NA	0.0405	[-0.0018, 0.0017]
Laplace-ML	-0.0004	NA	0.0415	[-0.0022, 0.0014]
ZERO-REML	-0.0064†	NA	0.0444	[-0.0083, -0.0045]
EBLUP-REML	0.0003	NA	0.0437	[-0.0016, 0.0023]
FELA-EM-REML	0.0009	NA	0.0452	[-0.0010, 0.0029]
$D_{22}^{(2)} = 0.3$				
ZERO-ML	0.2219†	-26.03	0.1353	[0.2170, 0.2268]
EBLUP-ML	0.2723†	-9.23	0.1386	[0.2663, 0.2783]
Laplace-ML	0.2785†	-7.18	0.1436	[0.2722, 0.2847]
ZERO-REML	0.2371†	-20.97	0.1351	[0.2318, 0.2423]
EBLUP-REML	0.2902†	-3.28	0.1468	[0.2837, 0.2966]
FELA-EM-REML	0.3006	0.21	0.1558	[0.2938, 0.3075]
$\sigma^2 = 0.25$				
ZERO-ML	0.2580†	3.20	0.0129	[0.2576, 0.2584]
EBLUP-ML	0.2498	-0.06	0.0089	[0.2494, 0.2502]
Laplace-ML	0.2497	-0.10	0.0089	[0.2493, 0.2501]
ZERO-REML	0.2582†	3.26	0.0130	[0.2577, 0.2586]
EBLUP-REML	0.2500	0.00	0.0089	[0.2496, 0.2504]
FELA-EM-REML	0.2498	-0.07	0.0089	[0.2494, 0.2502]

† The 95% CI does not cover the true value of the parameter and the estimate is significantly biased at 5% level.

4.6 Discussion

The results of Section 4.5 indicate that the proposed FELA-EM-REML algorithm gives accurate and reliable estimation results for both fixed effects and variance-covariance components when used to approximate the log-likelihood function in the nonlinear mixed model with complicated random effects such as nonlinear mixed model with two crossed random effects. It generally produces either unbiased or close-to-unbiased (%Bias < 1) estimates for both the fixed effects and the variance-covariance components with the 95% confidence interval coverages similar to the nominal value for all the fixed effects. The main advantages of this approximation method are its combination of computational efficiency (computation is slightly more than that of ML Laplace approximation yet less than that of REML Laplace approximation) and estimation accuracy (having an error of order $O(1/n^2)$ compared with the error of order $O(1/n)$ from Laplace approximation for estimating variance-covariance components).

For the fixed effects estimation, the estimates obtained from FELA-EM-REML approximation are similar to those from the Laplace-ML approximation, but are more accurate than those from the four linearization methods proposed by Wolfinger and Lin (1997). Both FELA-EM-REML and Laplace-ML approximations produce unbiased or close-to-unbiased (%Bias < 1%) estimates with the 95% confidence interval coverages similar to the nominal value for all fixed effects while the four linearization methods frequently generate significantly biased estimates with uncertain direction (i.e., the bias can be either negative or positive) and provide considerably lower coverage rates compared to the nominal one. We also observe that the two eblup-expansion

approximations generally produce smaller bias than the two zero-expansion approximations and the two zero-expansion approximations can give estimates with %Bias more than 18% (both zero-expansion approximations for estimating β_2 in the first-order compartment model). The RMSE of the fixed effects is generally similar for all approximations considered except for estimation of β_3 where the RMSE obtained from the two zero-expansion approximations is about twice as large as that from the other four approximations.

For the variance-covariance estimation, our proposed FELA-EM-REML algorithm is the only approximation that always gives unbiased estimates for variance and covariance parameters while the EBLUP-REML approximation ranks a second. The three ML approximations (ZERO-ML, EBLUP-ML, and Laplace-ML) generally underestimate the variance parameters while their performance for covariance parameters is uncertain. They can generate either unbiased or significantly biased estimates. The two zero-expansion approximations (ZERO-ML and ZERO-REML) can generate very poor estimates with %Bias more than 20% (i.e., estimating $D_{22}^{(1)}$ and $D_{22}^{(2)}$ in the first-order compartment model). The RMSE of the variance-covariance components is relatively similar for all approximations considered.

4.7 Conclusions

The proposed FELA-EM algorithm gives accurate and reliable estimation results, either unbiased or close-to-unbiased (%Bias < 1), with the 95% confidence interval coverages similar to the nominal value for both fixed effects and variance-covariance components when used to approximate the log-likelihood function in the nonlinear mixed model with complicated random effects such as nonlinear mixed model with two crossed random effects. For the fixed effects estimation, the Laplace approximation also produces unbiased or close-to-unbiased estimates with the observed coverages similar to the nominal 95% level while the four linearization methods frequently generate significantly biased estimates with uncertain direction and give significantly lower 95% confidence interval coverages than the nominal one. The two zero-expansion approximations can give estimates with %Bias more than 18% when estimating β_2 in the first-order compartment model. For the variance-covariance estimation, our proposed FELA-EM-REML algorithm is the only approximation that always gives unbiased estimates while the EBLUP-REML approximation ranks a second. The three ML approximations generally underestimate the variance parameters and can give either unbiased or significantly biased estimates for covariance. The two zero-expansion approximations can generate very poor estimates with %Bias more than 20%.

4.8 Summary

In this paper we have extended the FELA-EM algorithm for single level nonlinear mixed models to that for multilevel nonlinear mixed models with two crossed random effects. The extended FELA-EM algorithm is computationally efficient (similar to the REML version of the Laplace approximation) and shows great accuracy of the parameter estimation for multilevel level nonlinear mixed models with two crossed random effects. Two simulation studies were conducted to evaluate the accuracy of the extended the FELA-EM algorithm for estimating multilevel nonlinear mixed models with two crossed random effects and compare it with the ML version of the Laplace approximation and the four linearization methods proposed by Wolfinger and Lin (1997). Of all the approximation methods considered in this paper, the extended FELA-EM algorithm is the only one that gives unbiased or close-to-unbiased (%Bias < 1%) estimates for both the fixed effects and variance-covariance components and 95% confidence interval coverages similar to the nominal value for all the fixed effects. While the extended FELA-EM algorithm is computationally more intensive than the linearization methods, the sample code written in R and provided in the appendix of this dissertation is highly efficient and generally converges reliably and rapidly.

4.9 References

Bates, D., Maechler, M., Dai, B., 2008. Lme4: Linear mixed-effects models using S4 classes. R package version 0.999375-22. http://lme4.r-forge.r-project.org

Beal, S.L., Sheiner, L.B., 1992. NONMEM Users Guide. NONMEM Project Group, University of California, San Francisco, CA.

Davidian, M., Giltinan, D.M., 2003. Nonlinear Models for Repeated Measurement Data: An Overview and Update. *Journal of Agricultural, Biological, and Environmental Statistics* 8: 387-419.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society*, Series B 39: 1-38.

Ge, Z., Bickel, J.P., Rice A.J., 2004. An Approximate Likelihood Approach to Nonlinear Mixed Effects Models via Spline Approximation. *Computational Statistics & Data Analysis* 46(4): 747-776.

Hartley, H.O., Rao, J.N.K., 1967. Maximum likelihood estimation for the mixed analysis of variance model. *Biometrika* 54: 93-108.

Harville, D.A., 1977. Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. *Journal of the American Statistical Association* 72: 320-340.

Insightful Corporation, 2007. S-PLUS 8 for Windows Guide to Statistics. Seattle, WA, Insightful Corporation.

Jaffrezic, F., Meza, C., Lavielle, M., Foulley, M., 2006. Genetic analysis of growth curves using the SAEM algorithm. *Genetics Selection Evolution* 38: 583–600.

Jamshidian, M., Jennrich, R.I., 1997. Acceleration of the EM Algorithm by using Quasi-Newton Methods. *Journal of the Royal Statistical Society*, Series B 59: 569-587.

Kuhn, E., Lavielle, M., 2005. Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics & Data Analysis* 49: 1020-1038.

Leonard, T., Hsu, J.S.J., Tsui, K.W., 1989. Bayesian Marginal Inference. *Journal of the American Statistical Association* 84: 1051-1058.

Lindstrom, M.J., Bates, D.M., 1990. Nonlinear Mixed Effects Models for Repeated Measures Data. *Biometrics* 46: 673-687.

Millar, R.B., 2004. Simulated Maximum Likelihood Applied to Non-Gaussian and Nonlinear Mixed Effects and State-Space Models. *Australian & New Zealand Journal of Statistics* 46(4): 543-554.

Panhard, X., Samson, A., 2009. Extension of the SAEM algorithm for nonlinear mixed models with 2 levels of random effects. *Biostatistics* 10: 121-135.

Pinheiro, J.C., Bates, D.M., 1995. Approximations to the Log-likelihood Function in the Nonlinear Mixed-Effects Model. *Journal of Computational and Graphical Statistics* 4: 12-35.

Pinheiro, J.C., Bates, D.M., 2000. Mixed-Effects Models in S and S-PLUS. New York: Springer-Verlag.

R Development Core Team, 2009. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org

Ratkowsky, D.A., 1983. Nonlinear regression modeling: a unified practical approach. Marcel Dekker, NY.

SAS Institute, 2004. SAS OnlineDoc, Version 9.1.3. Cary, NC: SAS Institute Inc.

Tierney, L., Kadane, J.B., 1986. Accurate Approximations for Posterior Moments and Densities. *Journal of the American Statistical Association* 81(393): 82-86.

Tierney, L., Kass, R.E., Kadane, J.B., 1989. Fully exponential Laplace Approximations to Expectations and Variances of Nonpositive Functions. *Journal of the American Statistical Association* 84: 710-716.

Vonesh, E.F., 1996. A Note on the Use of Laplace's Approximation for Nonlinear Mixed-Effects Models. *Biometrika* 83: 447-452.

Walker, S.G., 1996. An EM Algorithm for Nonlinear Random Effects Models. *Biometrics* 52: 934-944.

Wang, J., 2007. EM algorithms for nonlinear mixed effects models. *Computational Statistics & Data Analysis* 51, 3244-3256.

Wolfinger, R.D., Lin, X., 1997. Two Taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis* 25: 465-490.

CHAPTER 5

MULTILEVEL NONLINEAR MIXED EFFECTS MODELS WITH BOTH CROSSED AND NESTED RANDOM EFFECTS APPLIED IN A REPLICATED LATIN SQUARE DESIGN FOR MODELING TEMPERATURE OF FEEDING PIGS

5.0 Abstract

A multilevel nonlinear mixed-effects model with both crossed and nested random effects applied in a replicated Latin square design is used to model feeding pigs' body temperature in conjunction with three different thermal environmental treatments, the amount of feed intake and the duration of the meal. Three-level random effects are introduced into a modified first order compartment model and the within-event correlation is described by an AR(1) model. We found that the thermal environmental treatments (28ºC + High air speed) and (18ºC + Low air speed) are significantly different from the reference treatment (28ºC + Low air speed) at the 5% level. The significant effects of feed intake and meal duration on feeding pigs' dynamic overall heat transfer coefficients such as the heat accumulation rate constant and the heat elimination rate constant were also detected. The nonlinear mixed-effects model was fit by the fully exponential Laplace approximation EM (FELA-EM-REML) algorithm, a newly developed method that can produce highly accurate estimates for variance-covariance components (giving an error of order $O(1/n^2)$).

Key Words: Nonlinear mixed models; EM algorithm; Fully exponential Laplace approximation; Crossed and nested random effects; Compartment model; Replicated Latin squares.

5.1 Introduction

An animal's ability to convert feed to weight gain is influenced by the thermal environment. A better understanding of an animal's response to the thermal environment can be achieved through investigating its body temperature data. By estimating an animal's dynamic overall heat transfer coefficients, such as the heat accumulation rate constant and the heat elimination rate constant using its body temperature data, we can help producers define an optimum range for the thermal environment so that they can adjust their production facilities to the environment best suited to enhance an animal's well being and feed efficiency. There are two objectives for this study. First, we fit a modified first-order compartment model to characterize the thermoregulatory responses of pigs during a feeding event. Parameters include the initial tympanic temperature, the heat accumulation rate constant and the heat elimination rate constant. Second, we compare those responses for three thermal environments (28°C air temperature and low air speed, 28°C air temperature and high air speed, and 18°C air temperature and low air speed) applied in a replicated Latin Square design and test the effect of the amount of feed intake as well as the effect of the meal duration.

Nonlinear mixed models discussed in this paper can contain both crossed and nested random effects. Currently, fitting nonlinear mixed-effects models with both crossed and nested random effects is a challenging topic in statistics. Although a number of software packages have been developed to fit nonlinear mixed models and generalized linear mixed models, most of them such as the SAS NLMIXED procedure (SAS Institute, 2004), NONMEM (Beal and Sheiner, 1992) and the MIXOR family of programs (Hedeker and Gibbons, 1996) only apply to single-level nonlinear mixed models and/or generalized linear mixed models without nested and crossed random effects. The NLME package developed by Pinheiro and Bates (2000), available in both R (R development core team 2009) and S-PLUS (Insightful Corporation 2007), is powerful for fitting nonlinear mixed models with nested random effects, but it does not fit nonlinear mixed models with crossed random effects. Rasbash and Goldstein (1994) show how to fit a linear mixed model with crossed random effects as a purely hierarchical formulation of nested random effects. Zhou et al. (2006) developed a method to enable NLME in R to fit a nonlinear mixed-effects model with crossed random effects based on Rasbash and Goldstein's idea for linear mixed-effects models. This method can be used to fit a nonlinear mixed effects model with arbitrary levels of crossed and nested random effects. The evaluation of the log-likelihood function in nonlinear mixed models is a rather complex numerical issue even for single-level nonlinear mixed models because it involves the evaluation of a multiple integral that, in most cases, does not have a closed-form expression. The only popular software package offering the capacity to fit nonlinear mixed models with both crossed and nested random effects is the SAS macro NLINMIX (Wolfinger and Lin, 1997). The NLINMIX macro is based on the

linearization methods. It uses a first-order Taylor series expansion to approximate the nonlinear model function around the current estimates of the fixed effects and a choice of expansion locus for the random effects – either around zero that is the expected value of the random effects (Wolfinger and Lin, 1997, ZERO-expansion method), or around the current estimates of the random effects (Wolfinger and Lin, 1997, EBLUP-expansion method), and then maximizes the likelihood corresponding to the resulting approximate linear mixed effects model. Linearization methods are computationally simple because they avoid complicated numerical integrations; however, they may produce substantial bias in parameter estimation with limited number of observations per subject and large variability of random effects (Ge, Bickel and Rice, 2004).

All nonlinear mixed models presented in this paper are fitted by the FELA-EM-REML algorithm, a newly developed method that can produce highly accurate estimates for variance-covariance components. A brief introduction to FELA-EM-REML for single-level nonlinear mixed models is given in Section 5.2.5. For theory and computation details of the FELA-EM-REML algorithm, see Chapters 3 and 4 for single- and multi-level nonlinear mixed models, respectively. The rest of this paper is organized as follows. In Section 5.2, we describe the experimental design of the data and address in detail the approach to nonlinear crossed random effects model building. In Section 5.3, we identify the significant random effects and examine the treatment effects as well as the effects of the amount of feed consumed and the duration of the meal. In Section 5.4, we describe the dynamic patterns of tympanic temperature. Finally, in Section 5.5, we summarize the study.

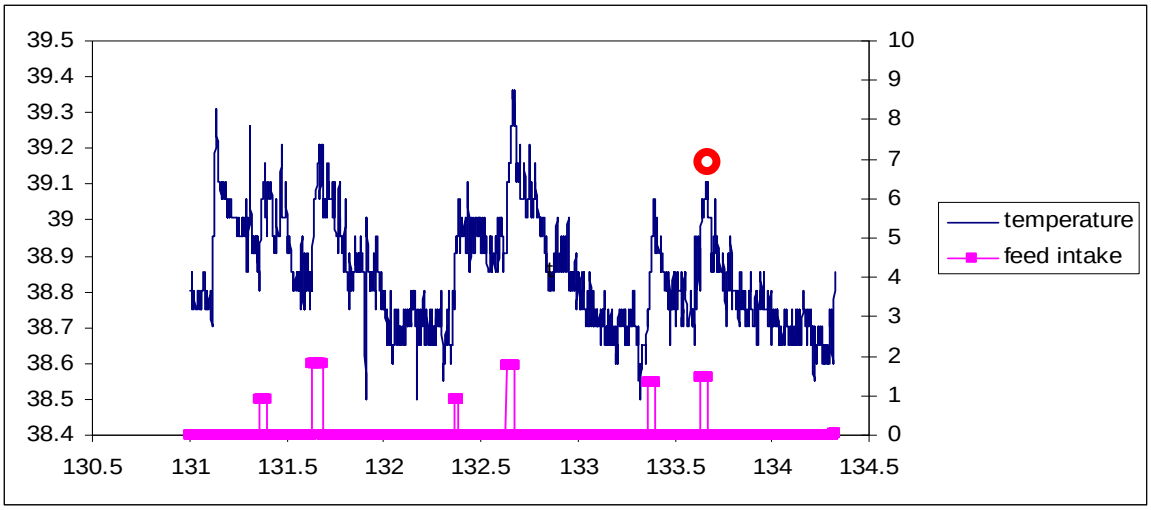
5.2 Materials and methods

5.2.1 Data

Eigenberg (1994) conducted an experiment to study the tympanic temperature of feeding pigs in response to three predefined thermal conditions. The experiment was designed as a Latin Square with three treatments, three pigs, and three treatment periods that are about three days in length. The treatments consisted of three combinations of ambient temperature and air speed. For the reference environment, treatment 1, the ambient temperature was set to 28 C and air speed was set to low (20 cm/s). Pigs housed in this environment are expected to be at rest for much of the time, and thus, generate a relatively stable body temperature record. For treatment 2, the air temperature was set to 28 C and air speed was set to high (90 cm/s). Treatment 3 completes the treatment group with air temperature set to 18 C and air speed set to low (20 cm/s). Both treatments 2 and 3 would be expected to produce higher thermal loss on the pig than treatment 1. Treatment 2 has higher thermal loss due to higher air speed and treatment 3 has higher thermal loss due to lower air temperature. Six pigs were randomly selected from eleven litters and they were split into two weight groups: three heavy animals (29.5±1.8 kg) and three light animals (22.5±1.0 kg). The heavier animals were exposed to the treatments first, then the lighter animals. Each weight group was used twice producing a total of four Latin Squares (two with heavy animals and two with light animals). During the experiment, each pig had the opportunity to eat approximately three meals every day for three days and each of the meals had the potential to produce one set of thermal index values such as the initial tympanic temperature, the heat accumulation rate constant, and

the heat elimination rate constant. The access to feed was controlled by solenoid latches on the feeding system. The pigs had access to feed only three times per day for a one-hour period. The meal times were: 2:00 AM, 8:30 AM and 3:00 PM. The tympanic temperature and feed intake of each pig were recorded every 48 seconds. An example showing changes in tympanic temperature and feed intake is presented in Figure 5.1. In this example, pig 27 (a member of the heavy group) was observed during the first experimental period where the second treatment (28ºC + High air speed) was applied. During this period, there were six feeding events and each feeding event produced a tympanic temperature spike. The whole study is a replicated Latin square design with three treatments, three pigs, and three treatment periods in each of the four squares. We limit our discussion to modeling body temperature for the heavy group but believe ideas can be extended to models for light group, and leave consideration of both groups for future research efforts. Only feeding events of the first largest meal on the third day of each period were included in the study. For each feeding event, the temperature record is analyzed for a record length of 80 minutes. In total, there are 18 feeding events considered in this study corresponding to the treatment structure presented in Table 5.1.

Figure 5.1: Example of changes in tympanic temperature (ºC) and feed intake (kg) of pigs over Julian calendar time for pig 27 (a member of the heavy group) during first experimental period in the second run under treatment 2 (28ºC and high air speed)



indicates that the feeding event was used in the study.

Table 5.1: Treatment structure for the replicated Latin square design

Pig No.		**Heavy Group**		
		85	**27**	**59**
First Run (Square 1)	Period 1	T2	T3	T1
	Period 2	T1	T2	T3
	Period 3	T3	T1	T2
Second Run (Square 2)	Period 1	T1	T2	T3
	Period 2	T3	T1	T2
	Period 3	T2	T3	T1

5.2.2 Statistical model

Compartment models are nonlinear models in which the response is described by a linear system of ordinary differential equations. Compartment models have been widely used in the literature for characterizing patterns of growth and decline. For examples see Bates and Watts (1988), Davidian and Giltinan (1995), Lindsey (1999), and Pinheiro and Bates (2000). If we assume that the change of body heat H in a feeding pig follows a one compartment model with first-order heat accumulation Ka and first-order heat elimination Ke during feed intake, the following differential equation can be created:

$$\frac{\partial \mathrm{Ha}}{\partial \mathrm{X}} = -\mathrm{Ka} \cdot \mathrm{Ha},$$

$$\frac{\partial \mathrm{H}}{\partial \mathrm{X}} = \mathrm{Ka} \cdot \mathrm{Ha} - \mathrm{Ke} \cdot \mathrm{H},$$

where Ha is heat produced by different activities such as standing up, moving, chewing, and digesting food. Integrating the pair of differential equations with initial conditions gives a modified three-parameter first-order compartment model:

$$\mathrm{Y} = \mathrm{Yo} + \frac{\mathrm{Ka}}{\mathrm{Ka} - \mathrm{Ke}}(\mathrm{e}^{-\mathrm{Ke}\cdot\mathrm{X}} - \mathrm{e}^{-\mathrm{Ka}\cdot\mathrm{X}}) + \varepsilon, \tag{5.1}$$

where the response variable Y is the tympanic (inner ear) temperature (C), the independent variable X is the time in hours, and the within-group errors ε are assumed to be normally distributed with mean 0 and variance-covariance matrix Ψ. There are three parameters in the model: Yo is the initial tympanic temperature (C), Ka is the heat accumulation rate constant (hour^{-1}), and Ke is the heat elimination rate constant (hour^{-1}). The heat accumulation rate constant Ka is a measure of the rate of increase in the body

temperature caused by the feeding event; while the heat elimination rate constant Ke is the rate of decrease in the body temperature. The larger Ka, the faster the body temperature approaches its maximum; the larger Ke, the faster the body temperature goes back to its initial value. For the nonlinear mixed-effects model applied in the replicated Latin squares with both crossed and nested random effects, two factors (three treatment levels and two squares), two covariates (feed intake and meal duration), and three levels of random effects were incorporated in the modified three-parameter first-order compartment model (5.1) for each of the three parameters:

$$
\begin{aligned}
&\mathrm{Yo} = \left[1 \mid C_1 \mid C_2 \mid 0 \mid 0 \mid C_5\right]\beta_1 + Zb_1,\\
&\mathrm{Ka} = \left[1 \mid C_1 \mid C_2 \mid C_3 \mid C_4 \mid C_5\right]\beta_2 + Zb_2,\\
&\mathrm{Ke} = \left[1 \mid C_1 \mid C_2 \mid C_3 \mid C_4 \mid C_5\right]\beta_3 + Zb_3,\\
&\beta_\ell = \begin{bmatrix} \beta_{\ell 1}\\ \beta_{\ell 2}\\ \beta_{\ell 3}\\ \beta_{\ell 4}\\ \beta_{\ell 5}\\ \beta_{\ell 6}\end{bmatrix}, \qquad \ell = 1,2,3, \qquad Z = \text{matrix of 1's},\\
&b_\ell = \begin{bmatrix} b_{\mathrm{PIG}i\ell}\\ b_{\mathrm{PD}j\ell}\\ b_{\mathrm{EVT}k\ell}\end{bmatrix}, \quad \begin{aligned} &(b_{\mathrm{PIG}i1}, b_{\mathrm{PIG}i2}, b_{\mathrm{PIG}i3})^{\mathrm{T}} \sim \text{i.i.d.}\, N(0, D_{\mathrm{PIG}}),\ i = 1,\ldots,6,\\ &(b_{\mathrm{PD}j1}, b_{\mathrm{PD}j2}, b_{\mathrm{PD}j3})^{\mathrm{T}} \sim \text{i.i.d.}\, N(0, D_{\mathrm{PD}}),\ j = 1,\ldots,6,\\ &(b_{\mathrm{EVT}k1}, b_{\mathrm{EVT}k2}, b_{\mathrm{EVT}k3})^{\mathrm{T}} \sim \text{i.i.d.}\, N(0, D_{\mathrm{EVT}}),\ k = 1,\ldots,18,\end{aligned}\\
&C_1 = \begin{cases} 1, & \text{Environment} = \text{Treatment\#2},\\ 0, & \text{else},\end{cases}\\
&C_2 = \begin{cases} 1, & \text{Environment} = \text{Treatment\#3},\\ 0, & \text{else},\end{cases}\\
&C_3 = \text{Amount of feed intake},\\
&C_4 = \text{Duration of the meal},\\
&C_5 = \begin{cases} 0, & \text{Square} = \text{First Run},\\ 1, & \text{Square} = \text{Second Run},\end{cases}
\end{aligned}
\tag{5.2}
$$

where the $\beta_{\lambda 1}$'s represent the means of the first treatment for Yo, Ka, and Ke corresponding to $\lambda = 1, 2, 3$, respectively; the $\beta_{\lambda 2}$'s represent the differences of the means between the second treatment and first treatment; the $\beta_{\lambda 3}$'s represent the differences of the means between the third and first treatments; the $\beta_{\lambda 4}$'s are the coefficients associated with the amount of feed intake; the $\beta_{\lambda 5}$'s are the coefficients related to the meal duration; and the $\beta_{\lambda 6}$'s are the coefficients related to the square. The random effects $b_{PIGi\lambda}$ represent the deviation from the population mean associated with the i^{th} pig for $i = 1, \ldots, 6$, similarly, random effects $b_{PDj\lambda}$ represent the deviations associated with the j^{th} period for $j = 1, \ldots, 6$, and random effects $b_{EVTk\lambda}$ represent the deviations associated with the k^{th} feeding event for $k = 1, \ldots, 18$ for each $\lambda = 1, 2, 3$. We further assume that b_{PIGi}, b_{PDj}, b_{EVTk} and ε are independent of each other. Since our preliminary analysis did not detect a significant treatment-by-square interaction effect on any of the three parameters, the treatment-by-square interaction was not included in the nonlinear mixed model (5.2).

5.2.3 Crossed and nested random effects

In our study, within a square, pig and period are crossed with each other and event is nested within the combination of pig and period. The random effects associated with pig and period are called crossed random effects and those associated with feeding event are nested random effects.

5.2.4 Model building

5.2.4.a) Random effects specification

We start with the multilevel nonlinear mixed effects model with all factors and covariates to investigate random-effects variation, which is a common model-building strategy used in statistical literatures for fitting mixed effects models (Littell, Milliken, Stroup, Wolfinger, and Schabenberger, 2006). In equation (5.1), the within-event errors, ε, are initially assumed to be independent $N(0,\sigma^2\mathbf{I})$, where $\mathbf{I}$ represents the identity matrix. In equation (5.2), all three parameters are initially considered to be mixed, including all factors, covariates, and pig-, period-, and event-level random effects. To avoid convergence problems, a diagonal structure of the variance-covariance matrices D_{PIG}, D_{PD}, and D_{EVT} is assumed. Under these assumptions, equations (5.1) and (5.2) are fit including all treatments, covariates, and random effects (all pig-, period-, and event-level random effects in all three parameters). Then, the random-effect terms with small variances are removed if the likelihood ratio test (LRT) is non-significant. In this way, a model is obtained including all significant random-effect terms under the assumptions of diagonal variance-covariance matrices of random effects and independence of within-event error with all treatments and covariates. Next, the variance-covariance matrices of random effects are changed from diagonal to unconstrained structure, and the model is refit. Again, the likelihood ratio test (LRT) is used to determine if the unconstrained covariance structure significantly improves the fit of the multilevel nonlinear mixed effects model.

5.2.4.b) Within-event error correlation structure specification

Since the tympanic temperature was collected over time for each eating event, we investigate the need for within-event correlation structures in the nonlinear mixed effects

model by looking at the plot of the empirical autocorrelation function (ACF) and assess the adequacy of a particular correlation structure by examining the ACF plot from the corresponding model.

5.2.4.c) Model diagnostics

After the random-effect and within-event error correlation structure specifications, the intrinsic relative curvature (IN) and the root mean square parameter effects curvature (PE) proposed by Bates and Watts (1980) are used to assess the nonlinear behavior of the final model by setting all random effects at zero (Noh and Lee 2008). When both IN and PE curvatures are less than or equal to 0.3, the nonlinearity of the final model is considered close-to-linear. The validity of the parameter estimates is examined by Box's approximate measure of bias (1972), percent excess variance based on Lowry and Morton's asymmetry measure (1983), and Hougaard's approximate measure of skewness (1985), which is suggested by Ratkowsky (1990). When %Bias and %excess variation are under 1%, and the absolute skewness is less than 0.25, we conclude that the behavior of the associated parameter is reasonably close-to-linear (Ratkowsky). The plot of standardized residuals versus fitted values is used to examine departure from model assumptions. The normal Q-Q plots of the random effects and the within-event errors are used to investigate the normality of the random effects and the within-event errors, respectively. A final assessment of the adequacy of the nonlinear mixed effects model is provided by a plot of the augmented predictions. If diagnostics show that all the assumptions are satisfied, the model can then be used to compare treatments over time and test the effects of the covariates.

5.2.5 Review of FELA-EM algorithm

The fully exponential Laplace approximation EM algorithm (FELA-EM) is an EM algorithm (Dempster, Laird, and Rubin 1977) for obtaining restricted maximum likelihood (REML) estimates in nonlinear mixed effects models. In the E-step of the FELA-EM algorithm, the fully exponential Laplace method (Tierney and Kadane 1986; Tierney, Kass, and Kadane 1989) is used to approximate the conditional expectations of the complete data sufficient statistics. The main advantages of this newly developed approach are its combination of computational efficiency (preserving the numerically simplicity of Laplace approximation) and great estimation accuracy (giving an error of order $O(1/n^2)$ for estimating variance-covariance components). The model used here to introduce the FELA-EM algorithm is a single-level nonlinear mixed effects model given by:

$$y_{ij} = f(\boldsymbol{\beta}, \boldsymbol{b}_i) + \varepsilon_{ij}, \quad i = 1,...,M, \quad j = 1,...,n_i, \tag{5.3}$$

where y_{ij} is the jth observation on the ith subject, f is a nonlinear function, $\boldsymbol{\beta}$ is a p-dimensional vector of fixed effects, $\boldsymbol{b}_i$ is a q-dimensional random effects vector associated with the ith subject (not varying with j) and assumed i.i.d. $N(\mathbf{0}, \boldsymbol{D})$, ε_{ij} is the error and assumed i.i.d. $N(0, \sigma^2)$, M is the number of subjects, and n_i is the number observations on the ith subject. It is further assumed that b_i and ε_{ij} are independent. For the nonlinear mixed effects model (5.3), assume a flat prior for the fixed effects $\boldsymbol{\beta}$ (Wolfinger 1993) and consider $c(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b})$ and $c(\boldsymbol{\beta}, \boldsymbol{b})$ as the complete data and the

missing data, respectively. Let $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ represent the parameters for which REML estimates are required. The FELA-EM algorithm alternates between an expectation step (E-step) and a maximization step (M-step) that can be described as follows.

5.2.5.a) E-step

The E-step computes the conditional expectation of the complete data log-likelihood $\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta})$,

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(k)}) = \int \ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}) p(\boldsymbol{\beta}, \boldsymbol{b} \mid \boldsymbol{y}; \boldsymbol{\theta}^{(k)}) d\boldsymbol{\beta} d\boldsymbol{b}, \tag{5.4}$$

where $p(\boldsymbol{\beta}, \boldsymbol{b} \mid \boldsymbol{y}; \boldsymbol{\theta}^{(k)})$ is the density of the missing data $c(\boldsymbol{\beta}, \boldsymbol{b})$ conditional on the observed data $\boldsymbol{y}$ at $\boldsymbol{\theta}^{(k)}$ and

$$\ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}) = K - \frac{N}{2}\log(\sigma^2) - \sum_{i=1}^{M} \frac{\left\| \boldsymbol{y}_i - f(\boldsymbol{\beta}, \boldsymbol{b}_i) \right\|^2}{2\sigma^2} - \frac{M}{2}\log(|\boldsymbol{D}|) - \sum_{i=1}^{M} \frac{\boldsymbol{b}_i^T \boldsymbol{D}^{-1} \boldsymbol{b}_i}{2} \tag{5.5}$$

for some constant K and $N = \sum_{i=1}^{M} n_i$ is the total number of observations. Since the density of the complete data $c(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b})$ is from the exponential family, the E-step can be simplified to compute the expectations of the sufficient statistics of the complete data for σ^2 and $\boldsymbol{D}$ on $\boldsymbol{y}$ at $\boldsymbol{\theta}^{(k)}$, which are given by $R = \sum_{i=1}^{M} \left\| \boldsymbol{y}_i - f(\boldsymbol{\beta}, \boldsymbol{b}_i) \right\|^2 = \left\| \boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b}) \right\|^2$ and $S_{m,n} = \sum_{i=1}^{M} b_{i,m} b_{i,n} = \boldsymbol{b}_m' \boldsymbol{b}_n$ for $m, n = 1, ..., q$, respectively.

5.2.5.b) M-step

The M-step is to find $\theta^{(k+1)}$ by solving the equation

$$\frac{\partial}{\partial \boldsymbol{\theta}} Q(\boldsymbol{\theta} \left| \boldsymbol{\theta}^{(k)} \right.) = \frac{\partial}{\partial \boldsymbol{\theta}} \int \ell(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}) p(\boldsymbol{\beta}, \boldsymbol{b} \left| \boldsymbol{y}; \boldsymbol{\theta}^{(k)} \right.) d\boldsymbol{\beta} d\boldsymbol{b} = 0. \tag{5.6}$$

By allowing differentiation under the integral sign, the unique solution to (5.6) is given by

$$\{\sigma^{(k+1)}\}^2 = \frac{1}{N} E(R \left| \boldsymbol{y}; \boldsymbol{\theta}^{(k)} \right.), \tag{5.7}$$

and

$$D_{m,n}^{(k+1)} = \frac{1}{M} E(S_{m,n} \left| \boldsymbol{y}; \boldsymbol{\theta}^{(k)} \right.), \qquad m, n = 1, ..., q. \tag{5.8}$$

5.2.5.c) Fully Exponential Laplace Approximation

Equations (5.7) and (5.8) generally cannot be computed analytically because both $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ and $E(S_{m,n}|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ can be nonlinear in the fixed and random effects. We approximate equations (5.7) and (5.8) by the fully exponential Laplace approximation introduced by Tierney and Kadane (1986). The fully exponential Laplace approximation to the ratio of two related integrals is given by

$$\begin{aligned}\frac{\int g(\boldsymbol{\varphi})\exp\{n\ell(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}}{\int \exp\{n\ell(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}} &= \frac{\int \exp\{n\ell^*(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}}{\int \exp\{n\ell(\boldsymbol{\varphi})\}d\boldsymbol{\varphi}} \\ &\approx \left(\frac{\det\{-\partial^2\ell(\hat{\boldsymbol{\varphi}})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'\}}{\det\{-\partial^2\ell^*(\hat{\boldsymbol{\varphi}}^*)/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'\}}\right)^{1/2}\exp\{n\ell^*(\hat{\boldsymbol{\varphi}}^*)-n\ell(\hat{\boldsymbol{\varphi}})\},\end{aligned} \tag{5.9}$$

where $g(\boldsymbol{\varphi})$ is a positive scalar function, $\ell^*(\boldsymbol{\varphi}) = \log\{g(\boldsymbol{\varphi})\}/n + \ell(\boldsymbol{\varphi})$, and $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\varphi}}^*$ maximize ℓ and ℓ^*, respectively. Although the errors in the Laplace approximations to the two integrals (numerator and denominator) in (5.9) are of order $O(1/n)$, the error in the ratio (5.9) is of order $O(1/n^2)$ due to the cancellation of the similar error terms in the approximation to the two integrals. Tierney, Kass and Kadane (1989) generalize the Laplace approximation in (5.9) so that $g(\boldsymbol{\varphi})$ can take on negative values. Their approach is to first approximate the moment generating function of $g(\boldsymbol{\varphi})$ that is strictly positive, and then approximate the ratio in (5.9) by evaluating the first derivative of the approximation to the moment generating function at 0. The moment generating function of $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ is given by

$$M_R(t) = \frac{\int \exp\{t\|\boldsymbol{y} - f(\boldsymbol{\beta},\boldsymbol{b})\|^2 + \hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})\}d\boldsymbol{\beta}d\boldsymbol{b}}{\int \exp\{\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})\}d\boldsymbol{\beta}d\boldsymbol{b}}, \tag{5.10}$$

where

$$\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)}) = -\frac{1}{2\{\sigma^{(k)}\}^2}\sum_{i=1}^{M}\|\boldsymbol{y}_i - f(\boldsymbol{\beta},\boldsymbol{b}_i)\|^2 - \frac{1}{2}\sum_{i=1}^{M}\boldsymbol{b}_i^T\{\boldsymbol{D}^{(k)}\}^{-1}\boldsymbol{b}_i.$$

Applying the fully exponential Laplace approximation (5.9) to (5.10) yields

$$\begin{aligned}\widetilde{M}_R(t) = &\left(\frac{\det[-\partial^2\hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{b}};\boldsymbol{\theta}^{(k)})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}{\det[-\partial^2\{t\|\boldsymbol{y} - f(\hat{\boldsymbol{\beta}}^*,\hat{\boldsymbol{b}}^*)\|^2 + \hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}}^*,\hat{\boldsymbol{b}}^*;\boldsymbol{\theta}^{(k)})\}/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}\right)^{1/2} \\ &\times\exp\{t\|\boldsymbol{y} - f(\hat{\boldsymbol{\beta}}^*,\hat{\boldsymbol{b}}^*)\|^2 + \hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}}^*,\hat{\boldsymbol{b}}^*;\boldsymbol{\theta}^{(k)}) - \hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{b}};\boldsymbol{\theta}^{(k)})\},\end{aligned} \tag{5.11}$$

where $\hat{\boldsymbol{\varphi}}^* = c(\hat{\boldsymbol{\beta}}^*,\hat{\boldsymbol{b}}^*)$ maximizes $t\|\boldsymbol{y} - f(\boldsymbol{\beta},\boldsymbol{b})\|^2 + \hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})$ and $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}},\hat{\boldsymbol{b}})$ maximizes $\hbar(\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{b};\boldsymbol{\theta}^{(k)})$. Then $E(R|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ can be approximated by $\partial\widetilde{M}_R(t)/\partial t$ evaluated at 0. Similarly, one can show that the fully exponential Laplace approximation to the moment generating function of $E(S_{m,n}|\boldsymbol{y};\boldsymbol{\theta}^{(k)})$ for $m,n = 1,...,q$ is given by

$$\begin{aligned}\widetilde{M}_{S_{m,n}}(t) = &\left(\frac{\det[-\partial^2\hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{b}};\boldsymbol{\theta}^{(k)})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}{\det[-\partial^2\{t\boldsymbol{b}_m'\boldsymbol{b}_n + \hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}}^{**},\hat{\boldsymbol{b}}^{**};\boldsymbol{\theta}^{(k)})\}/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}']}\right)^{1/2} \\ &\times\exp\{t\boldsymbol{b}_m'\boldsymbol{b}_n + \hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}}^{**},\hat{\boldsymbol{b}}^{**};\boldsymbol{\theta}^{(k)}) - \hbar(\boldsymbol{y},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{b}};\boldsymbol{\theta}^{(k)})\},\end{aligned} \tag{5.12}$$

where $\hat{\boldsymbol{\varphi}}^{**} = c(\hat{\boldsymbol{\beta}}^{**}, \hat{\boldsymbol{b}}^{**})$ maximizes $t\boldsymbol{b}_m'\boldsymbol{b}_n + \hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ and $\hat{\boldsymbol{\varphi}} = c(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{b}})$ maximizes $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$. As pointed by Tierney and Kadane (1986), once $\hat{\boldsymbol{\varphi}}$, the maximum of $\hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$, has been determined, it can be used as starting values to find $\hat{\boldsymbol{\varphi}}^{*}$ and $\hat{\boldsymbol{\varphi}}^{**}$, the maximum of $t\|\boldsymbol{y} - f(\boldsymbol{\beta}, \boldsymbol{b})\|^2 + \hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ and $t\boldsymbol{b}_m'\boldsymbol{b}_n + \hbar(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}; \boldsymbol{\theta}^{(k)})$ in (5.11) and (5.12), respectively. Generally, the number of iterations needed to find $\hat{\boldsymbol{\varphi}}^{*}$ and $\hat{\boldsymbol{\varphi}}^{**}$ from $\hat{\boldsymbol{\varphi}}$ is quite small and replacing $\hat{\boldsymbol{\varphi}}^{*}$ and $\hat{\boldsymbol{\varphi}}^{**}$ by two Newton steps from $\hat{\boldsymbol{\varphi}}$ are usually sufficient. Thus, the computational requirements of the fully exponential Laplace approximation are rather minimal. Once the estimates of variance and covariance parameters $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ are obtained, the fixed and random effects $\boldsymbol{\beta}$ and $\boldsymbol{b}$ can be estimated via the standard Laplace approximation (Pinherio and Bates 1995) to the observed data log-likelihood of Model (5.3) by holding $\boldsymbol{\theta} = c(\sigma^2, \boldsymbol{D})$ at $\hat{\boldsymbol{\theta}}$, the estimates at convergence from the FELA-EM algorithm.

5.3 Results and discussion

5.3.1 Specification of random effects

We start with the multilevel nonlinear mixed effects model with all treatments and covariates to investigate random-effects variation. Diagonal structures of the variance-covariance matrices are initially assumed for D_{PIG}, D_{PD}, and D_{EVT} and we also assume the within-event errors, ε, to be independent. We remove one random effect term from the parameters at a time. This results in several models with different random-effects components (Table 5.2). Since the reduced models are nested within the full model and the same fixed-effects structures are used, LRTs can be used to check if the reduction in random effects causes any significant changes in model performance. Comparisons of the seven models are shown in Table 5.3. The similar Log-Likelihoods and the large p-values for the likelihood ratio test suggest that the seven models give essentially equivalent fits so the simpler model (i.e., the model with fewer random effects), Model 7, is preferred. The smallest AIC and BIC values further confirm that Model 7 has the best performance. That is to say, the pig- and period-level random effects associated with all three parameters (Yo, Ka, and Ke) can be safely dropped from the full model (Model 1). Starting with Model 7 (i.e., nonlinear mixed model with the diagonal D_{EVT}), we assume that the event-level random effects are correlated, which results in Model 8 (i.e., nonlinear mixed model with the unconstrained D_{EVT}). The fitting comparison of Model 7 with Model 8 indicates that the LRT is not significant at 5% level (p-value=.975). Therefore, Model 7 with diagonal variance-covariance matrices of the event-level random effects is preferable over Model 8 with unconstrained variance-covariance structures of random-effects.

Table 5.2: Variances for nonlinear mixed models with different random-effects components

Model	Pig			Period			Event		
	Yo	Ka	Ke	Yo	Ka	Ke	Yo	Ka	Ke
1	7.6e-6	0.0010	0.0008	1.9e-6	0.0770	0.1092	0.0256	0.4814	0.4837
2	1.7e-5	0.0006	6.7e-5		0.0665	0.1098	0.0256	0.4907	0.4848
3		0.0006	0.0007		0.0536	0.1091	0.0257	0.5010	0.4830
4			0.0010		0.0427	0.1010	0.0256	0.5085	0.4842
5					0.0326	0.1019	0.0256	0.5195	0.4874
6						0.1036	0.0258	0.5616	0.4863
7							0.0257	0.5580	0.5816

Table 5.3: Comparisons of model fit with different random-effects components

Model*	AIC	BIC	Log-Likelihood	LRT†	p-value
1	-2339.68	-2291.45	1179.838		
2	-2341.68	-2298.28	1179.838	<0.001	>0.9999
3	-2343.67	-2305.09	1179.836	0.004	0.9980
4	-2345.66	-2311.91	1179.830	0.016	0.9995
5	-2347.64	-2318.71	1179.819	0.038	0.9998
6	-2349.54	-2325.43	1179.771	0.134	0.9997
7	-2351.21	-2331.92	1179.605	0.466	0.9982

* Model as described in Table 2.

† Log-Likelihood Ratio Test is calculated with respect to Model 1.

5.3.2 Specification of within-event error correlation structure

We use the function **acf** in R package MASS (Venables and Ripley 2002) to investigate the within-event error correlation at different lags. The plot of the autocorrelation function of Model 7 shows high correlations among the within-event error (Figure 5.2). After the inclusion of the first-order autocorrelation structure AR(1) to Model 7 for modeling the within-event error correlation structure, which produces Model 9, we use the autocorrelation function again to investigate the correlation at different lags. The plot of the autocorrelation function is displayed in Figure 5.3 and does not show any high correlations among the within-event error. Hence, Model 9 is the preferred model for the comparison of the treatments and the test of the covariate effects.

Figure 5.2: Autocorrelation function corresponding to the within-event errors of Model 7

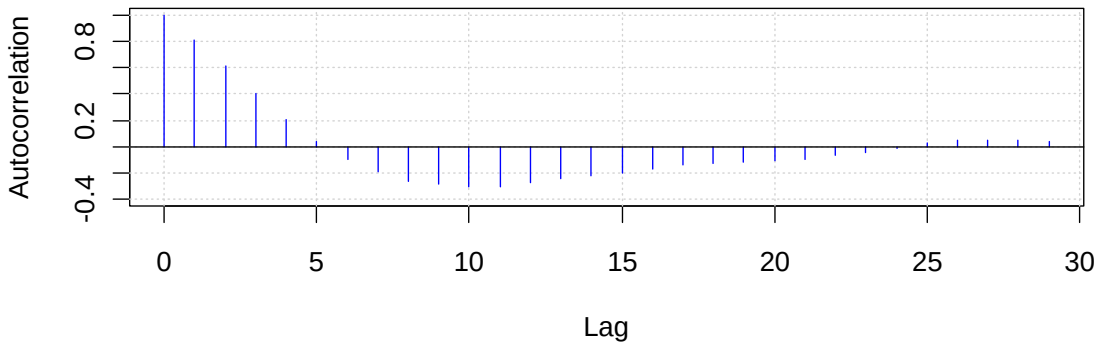
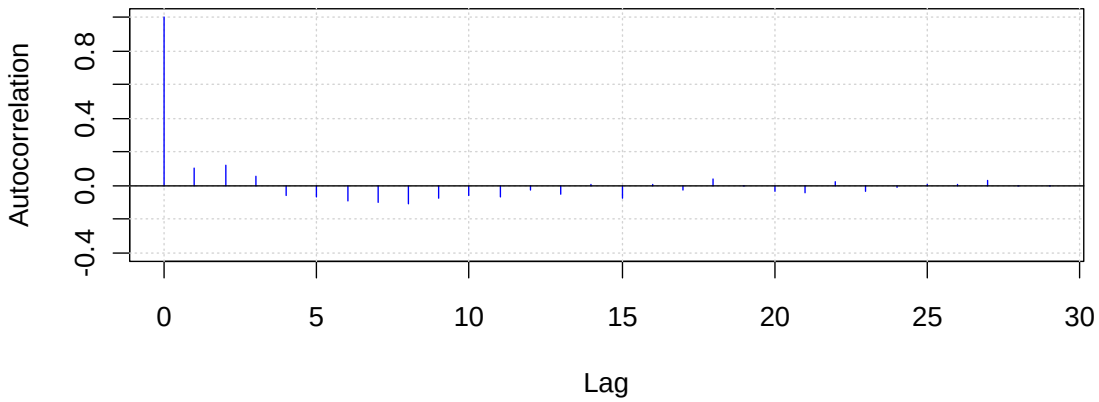


Figure 5.3: Autocorrelation function corresponding to the within-event errors of Model 9



5.3.3 Model diagnostics

The intrinsic relative curvature (IN) and the root mean square parameter effects curvature (PE) of Model 9 are 0.051 and 0.132, respectively. Since both IN and PE curvatures are less than 0.3 we may say that the nonlinearity of the final model (Model 9) is close-to-linear when all random effects equal zero. The asymptotic properties of fixed effects estimates in Model 9 are given in Table 5.4. The small %Bias (< 1%), %excess variation (< 1%), and the absolute skewness (< 0.25) indicate that the behavior of all fixed effects parameters are also reasonably close-to-linear. Figure 5.4, the plot of the standardized residuals versus the fitted values corresponding to Model 9 shows that the residuals are distributed symmetrically around zero, with an approximately constant variance. It does not indicate any departure from the model assumptions for the within-event errors, except for three possible outlying observations which are located outside ±4 standard deviations (Kutner, Nachtsheim, and Neter 2004). Similarly, the normal Q-Q plots of the standardized residuals (Figure 5.5) and the random effects (Figure 5.6) do not show any violations of the normality assumption for the within-event errors and the random effects, respectively. A final assessment of the adequacy of the nonlinear mixed-effects model is given by the plot of the augmented predictions in Figure 5.7. From the plot, we can see that the predicted temperatures are close to the observed values. Therefore, we conclude that the final nonlinear mixed-effects model (Model 9) provides a reasonable representation of the tympanic temperatures during feeding events.

Table 5.4: Asymptotic properties of fixed effects estimates in Model 9

Fixed Effects		Estimate	%Bias	%Excess Variance	Skewness
Yo	β_{11}: T1	39.0318	-0.0008	0.0961	-0.0409
	β_{12}: T2 – T1	0.0857	0.1894	0.1407	0.0629
	β_{13}: T3 – T1	-0.0519	-0.1999	0.1369	0.0654
	β_{14}: Square	-0.1877	-0.0272	0.1985	-0.0087
Ka	β_{21}: T1	4.7008	0.1211	0.1641	0.0767
	β_{22}: T2 – T1	-1.5280	0.1709	0.2797	-0.1176
	β_{23}: T3 – T1	-0.8912	0.2163	0.3286	-0.1158
	β_{24}: Feed Intake	1.1827	0.1448	0.1349	0.0678
	β_{25}: Meal Duration	-4.6128	0.0860	0.1264	-0.0494
	β_{26}: Square	0.4558	-0.0075	0.1463	-0.0130
Ke	β_{31}: T1	1.1596	0.0646	0.2365	0.1479
	β_{32}: T2 – T1	1.4578	0.0551	0.5887	0.2275
	β_{33}: T3 – T1	1.3631	0.1031	0.4311	0.2140
	β_{34}: Feed Intake	1.0154	0.1531	0.2876	0.1677
	β_{35}: Meal Duration	-1.7973	0.1880	0.2791	-0.1616
	β_{36}: Square	-0.1211	-0.1485	0.5338	-0.0127

Figure 5.4: Scatter plot of standardized residuals versus fitted values for Model 9

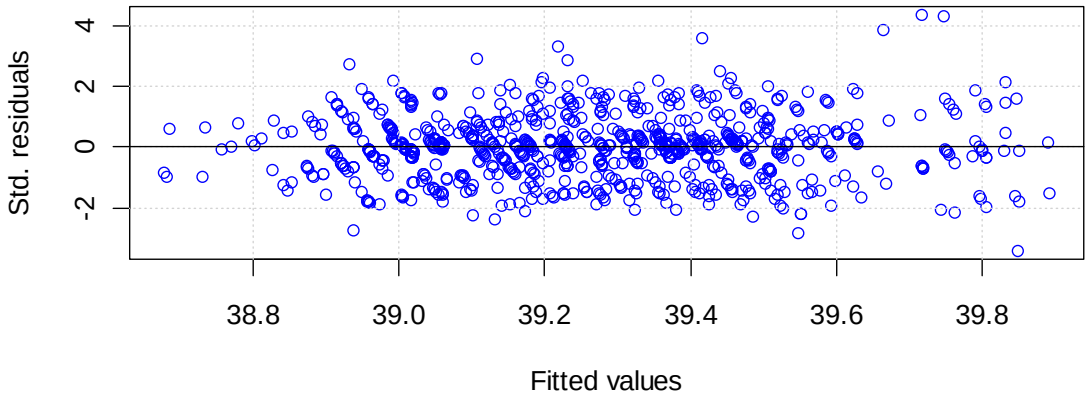


Figure 5.5: Normal Q-Q plot of standardized residuals for Model 9

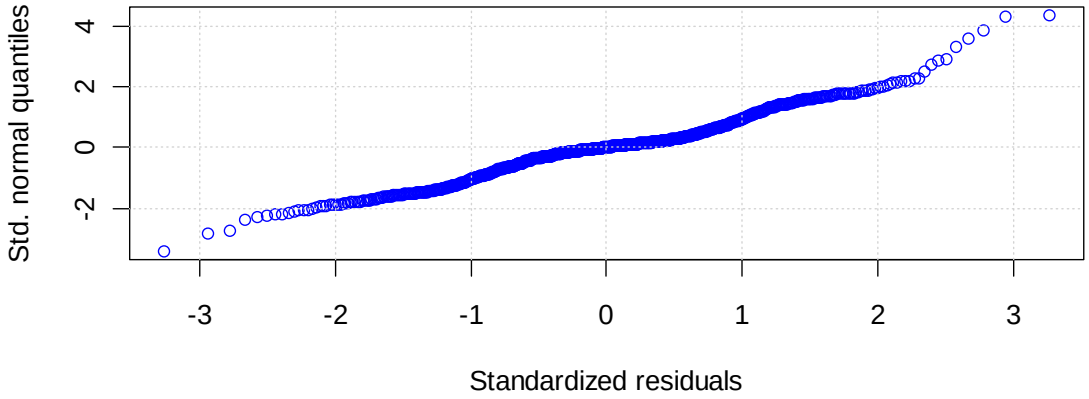


Figure 5.6: Normal Q-Q plots of the estimated random effects for Model 9

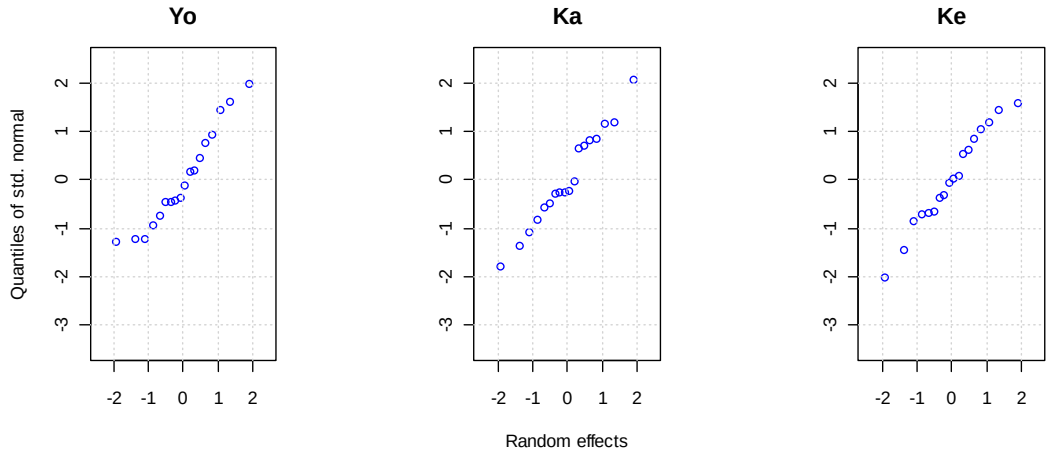
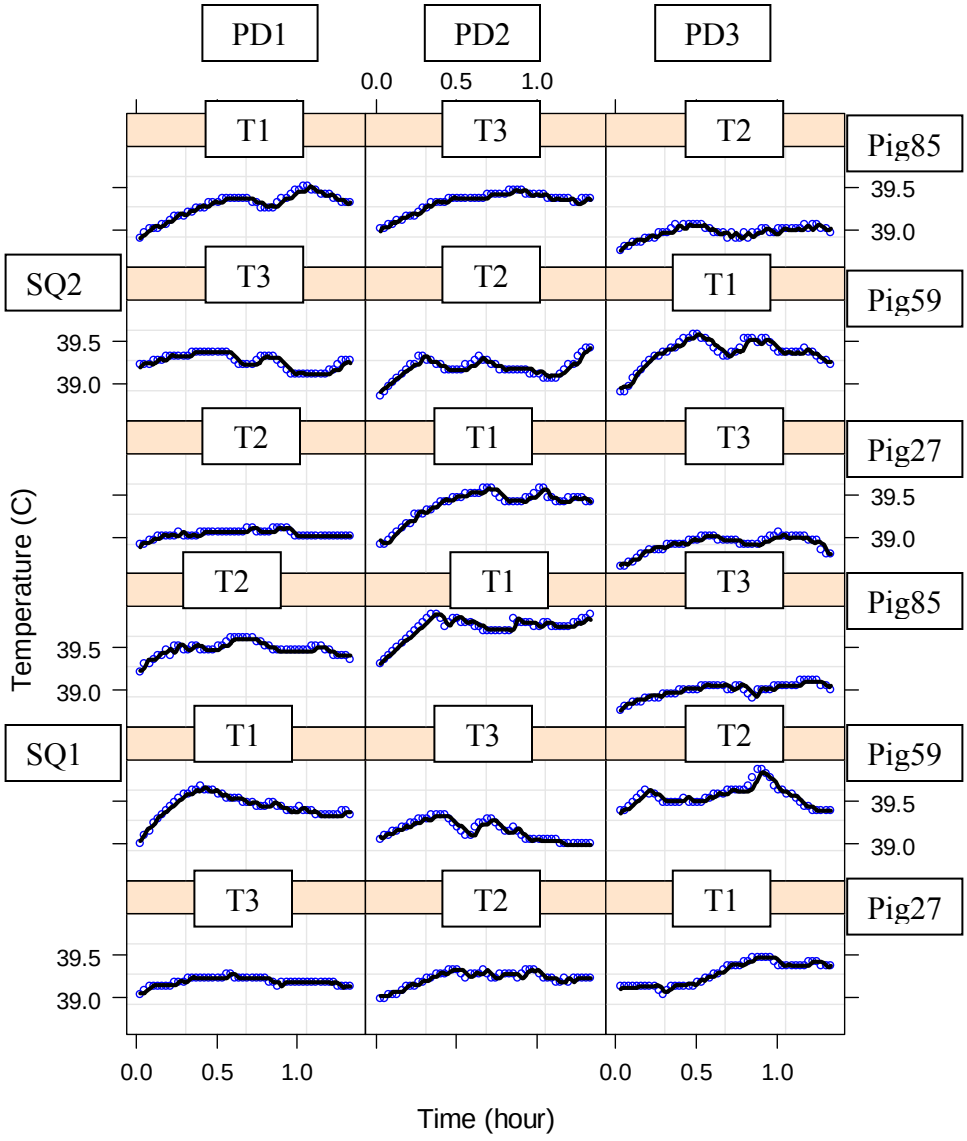


Figure 5.7: Observed (○) and predicted (—) tympanic temperatures (C) over time (min) for eighteen feeding events



5.3.4 Comparison of the Three Thermal Environmental Treatments and Test of the Feed Intake and Meal Duration Effects

We compared the three thermal environmental treatments and tested the effects of feed intake, meal duration, and square defined in Table 5.1 based on the results from Model 9 (Table 5.5). We found that treatment 2 (28ºC + High air speed) and treatment 3 (18ºC + Low air speed) were significantly different from treatment 1 (28ºC + Low air speed) for both accumulation and elimination rate constant parameters: Ka and Ke. The treatment effect on the initial tympanic temperature (Yo) was not detected. When testing the effects of feed intake and meal duration, we found that the amount of feed intake had no significant effect on Ka or Ke, while the meal duration had a significant effect on Ka, but not on Ke. The initial body temperature Yo in the two runs was significantly different, but not for Ka and Ke. From the parameter estimates, we found that both increasing the air speed and decreasing the environmental temperature could help pigs eliminate heat effectively. In comparison with the reference treatment 1, increasing the air speed (treatment 2) decreased the heat accumulation rate constant by 1.5280 $hour^{-1}$ and increased the heat elimination rate constant by 1.4578 $hour^{-1}$; while decreasing environmental temperature (treatment 3) decreased the heat accumulation rate constant by 0.8912 $hour^{-1}$ and also increased the heat elimination rate constant by 1.3631 $hour^{-1}$. Examination of the estimates of meal duration and square showed that increasing the meal duration was able to decrease the heat accumulation rate constant by 4.6126 $hour^{-1}$ while the average initial body temperature (Yo) in the second run was 0.1877 C lower than that in the first run. These results agree with the partial results obtained by Eigenberg (1994) and Zhou et al. (2006). Eigenberg's work (1994) focused on modeling

the index of heat dissipation and showed similar treatment differences, that is, both increasing the air speed and decreasing the environmental temperature can help pigs eliminate heat effectively. Zhou et al. (2006) investigated the feeding pigs' dynamics specifying two heat transfer rate coefficients using the data from the first run of the heavy group (Eigenberg, 1994). They found similar treatment effects but they did not include the effects of feed intake and meal duration in their model. The proposed model in this paper includes two rate coefficients (i.e., heat accumulation and heat elimination rate constants), three treatments levels and the two covariates (i.e., feed intake and meal duration). We detected the significant effects of treatments on both heat accumulation and heat elimination rate constants. We, also, found that the meal duration was significant. This indicates the quicker a pig eats the faster its body temperature approaches a maximum.

Table 5.5: Estimates of fixed effects coefficients for Model 9

Fixed Effects		Estimate	Standard Error	P-value
Yo	β_{11}: T1	39.0318	0.0780	< 0.0001
	β_{12}: T2 – T1	0.0857	0.0951	0.3602
	β_{13}: T3 – T1	-0.0519	0.0950	0.5732
	β_{14}: Square	-0.1877	0.0773	0.0149
Ka	β_{21}: T1	4.7008	1.0120	< 0.0001
	β_{22}: T2 – T1	-1.5280	0.4919	0.0019
	β_{23}: T3 – T1	-0.8912	0.4845	0.0645
	β_{24}: Feed Intake	1.1827	0.8440	0.1579
	β_{25}: Meal Duration	-4.6128	1.4234	0.0012
	β_{26}: Square	0.4558	0.6091	0.4452
Ke	β_{31}: T1	1.1596	0.9623	0.2236
	β_{32}: T2 – T1	1.4578	0.4823	0.0025
	β_{33}: T3 – T1	1.3631	0.4741	0.0040
	β_{34}: Feed Intake	1.0154	0.8621	0.2341
	β_{35}: Meal Duration	-1.7973	1.4195	0.2014
	β_{36}: Square	-0.1211	0.6291	0.8304

5.4 Conclusions

The proposed compartment model does a good job of fitting the tympanic temperatures of pigs during a feeding event. All three parameters, initial tympanic temperature (Yo), heat accumulation rate constant (Ka) and heat elimination rate constant (Ke), in the proposed model were considered as mixed effects. The event-level random effects influenced all three parameters Yo, Ka and Ke independently while the pig- and period-level random effects did not have influence on any of the parameters and hence can be safely dropped from the nonlinear mixed model. Both treatment 2 (28ºC + High air speed) and treatment 3 (18ºC + Low air speed) were significantly different from the reference treatment 1 (28ºC + Low air speed) for both Ka and Ke. The heat accumulation rate constant was lower in treatment 2 and treatment 3 than in treatment 1 while the heat elimination rate constant was greater in both treatment 2 and treatment 3 when compared with the reference treatment 1. The amount of feed intake had no significant influence on any of the three parameters (Yo, Ka, and Ke) while the length of meal duration significantly decreased Ka. There was no significant effect of meal duration on either the initial body temperature or the heat elimination rate constant. The initial body temperature Yo in the second run was significantly lower than that in the first run while the effect of square on either Ka or Ke was not detected.

5.5 Summary

This study provides a three-parameter modified first-order compartment model to describe the thermoregulatory responses of pigs during a feeding event. These thermoregulatory responses can best be described in terms of the initial tympanic temperature, the heat accumulation rated constant, the heat elimination rate constant and factors that affect them. The crossed and nested random effects have been introduced into the model to simultaneously model the pig, period, and event variations. Comparisons of three environment treatment effects over time and testing of the feed intake, meal duration, and square effects are incorporated in the proposed model. We present a general approach to building a multilevel nonlinear mixed model with both crossed and random effects and advocate in detail a way to simplify the random effect terms. We fit the nonlinear mixed effects model by the fully exponential Laplace approximation EM algorithm, a newly developed method that can give an error as small as of order $O(1/n^2)$ for variance-covariance parameters.

5.6 References

Bates, D.M. and Watts, D.L., 1980. Relative curvature measures of nonlinearity. *Journal of the Royal Statistical Society*, Series B 42, 1-25.

Bates, D.M. and Watts, D.G., 1988. Nonlinear regression analysis and its applications. Wiley: New York.

Beal, S.L., Sheiner, L.B., 1992. NONMEM User's Guides. NONMEM Project Group, University of California, San Francisco.

Box, M.J., 1972. Bias in nonlinear estimation. *Journal of the Royal Statistical Society*, Series B 33, 171-201.

Davidian, M. and Giltinan, D.M., 1995. Nonlinear models for repeated measurement data. Chapman and Hall: New York.

Dempster, A.P., Laird, N.M., and Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society*, Series B 39, 1-38.

Eigenberg, R.A., 1994. Tympanic Temperature Transient Response as an Index of Heat Dissipation in Swine. Ph.D. thesis, University of Nebraska - Lincoln.

Ge, Z., Bickel, J.P., Rice, A.J., 2004. An approximate likelihood approach to nonlinear mixed effects models via spline approximation. *Computational Statistics & Data Analysis* 46, 747-776.

Hedeker, D., Gibbons, R. D., 1996. MIXOR: A Computer Program for Mixed-Effects Ordinal Regression Analysis. *Biometrics* 50: 933-944.

Hougaard, P., 1985. The appropriateness of the asymptotic distribution in a nonlinear regression model. *Journal of the Royal Statistical Society*, Series B 47, 103-114.

Insightful Corporation, 2007. S-PLUS 8 for Windows Guide to Statistics. Seattle, WA, Insightful Corporation.

Lindsey, J.K., 1999. Nonlinear models in medical statistics. Oxford University Press.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., Schabenberger, O., 2006. SAS System for Mixed Models. Cary, NC: SAS Institute, Inc.

Lowry, R. and Morton, R., 1983. An asymmetry measure for estimators in non-linear regression models. Proceedings of the 44th Session of the International Statistical Institute, Madrid, Contributed Papers 1, 351-354.

Kutner, M.H., Nachtsheim, C.J., and Neter, J., 2004. Applied Linear Regression Models (Fourth edition). McGraw-Hill/Irwin: New York.

Noh, M. and Lee, Y., 2008. Hierarchical-likelihood approach for nonlinear mixed-effects models. *Computational Statistics & Data Analysis* 52, 3517-3527.

Pinheiro, J.C. and Bates, D.M., 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4, 12-35.

Pinheiro, J.C. and Bates, D.M., 2000. Mixed-Effects Models in S and S-PLUS. Springer: New York.

R Development Core Team, 2009. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org

Rasbash, J., Goldstein, H., 1994. Efficient Analysis of Mixed Hierarchical and Cross-Classified Random Structures Using a Multilevel Model. *Journal of Educational and Behavioural Statistics* 19: 337-350.

Ratkowsky, D., 1990. Handbook of nonlinear regression models. Marcel Dekker: New York and Basel.

SAS Institute, 2004. SAS OnlineDoc, Version 9.1.3, Cary, NC: SAS Institute Inc.

Tierney, L. and Kadane, J.B., 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81, 82-86.

Tierney, L., Kass, R.E., and Kadane, J.B., 1989. Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association* 84, 710-716.

Venables, W.N. and Ripley, B.D., 2002. Modern Applied Statistics with S (Fourth edition). Springer.

Wolfinger, R., 1993. Laplace's approximation for nonlinear mixed models. *Biometrika* 80, 791-795.

Wolfinger, R.D., Lin, X., 1997. Two Taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis* 25, 465-490.

Zhou, M., Parkhurst, A.M., Eigenberg, R.A., Nienaber, J.A. and Hahn, G.L. 2006. Evaluating nonlinear crossed random effects models for comparing temperature of eating pigs under different thermal environments. *Proceedings of Eighteenth Annual Kansas State University Conference on Applied Statistics in Agriculture Proceedings.*

APPENDIX: R program for fitting the logistic model formulated by the equation (4.4.1) using FELA-EM algorithm

```
#######################
# GENERATE A DATA SET #
#######################

library(MASS) # load MASS library

n1 <- 10 # levels of CF1
n2 <- 10 # levels of CF2
n <- 7 # number of obs for each individual
nP <- 3 # number of parameters

CF1 <- gl(n1, n2*n) # create CF1
CF2 <- gl(n2, n, n2*n*n1) # create CF2

a <- c(200, 700, 350) # fixed effects
a.ini <- a
s2 <- 625 # variance of error
s2.ini <- s2
D1 <- matrix(c(2500,-1875, -1875,15625), ncol = 2) # D matrix of CF1
D1.ini <- D1
D2 <- matrix(c(2600,-1950, -1950,16250), ncol = 2) # D matrix of CF2
D2.ini <- D2

b1 <- mvrnorm(n1, integer(2), D1) # random effects of CF1
b1.1 <- rep(b1[,1], each = n2*n) # b1 in parameter 1
b1.2 <- rep(b1[,2], each = n2*n) # b1 in parameter 2

b2 <- mvrnorm(n2, integer(2), D2) # random effects of CF2
b2.1 <- rep(b2[,1], each = n, times = n1) # b2 in parameter 1
b2.2 <- rep(b2[,2], each = n, times = n1) # b2 in parameter 2

error <- rnorm(n1*n2*n, mean = 0, sd = sqrt(s2)) # error

phi1 <- a[1]  + b1.1 + b2.1
phi2 <- a[2]  + b1.2 + b2.2
phi3 <- a[3]

x <- rep(c(118,484,664,1004,1231,1372,1582), len=n*n1*n2) # predictor
y <- phi1/(1 + exp(-(x - phi2)/phi3)) + error # response
DATA <- data.frame(CF1, CF2, x, y) # create data

# remove all variables
rm(n1, n2, n, nP, CF1, CF2, a, s2, D1, D2)
rm(b1, b1.1, b1.2, b2, b2.1, b2.2)
rm(error, phi1, phi2, phi3, x, y)
```

```
########################################
# EXTRACT USEFUL INFORMATION FROM DATA #
########################################

CF1 <- DATA$CF1
CF2 <- DATA$CF2
x <- DATA$x
y <- DATA$y
n1 <- length(levels(CF1))
n2 <- length(levels(CF2))
N <- length(y) # number of total obs
nP <- 3 # number of parameters in the model
nb <- 2*n1 + 2*n2 # number of random effects


######################
# DEFINE EXPRESSIONS #
######################

model.exp <- expression(phi1/(1 + exp(-(x - phi2)/phi3))) # model
der.exp <- deriv3(model.exp, c("phi1", "phi2", "phi3")) # derivative


########################
# START OF nllike.obj #
########################

nllike.obj <- function(ab) {
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) #f value
  grad <- attr(der, "gradient") #1st derivative
  hess <- attr(der, "hessian") #2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2  + sum(b2 %*% invD2 * b2)/2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <- rowsum(grad[,1]*grad[,1],CF1)/s2 + invD1[1,1]
  diag(G.b1.12) <- rowsum(grad[,1]*grad[,2],CF1)/s2 + invD1[1,2]
  diag(G.b1.22) <- rowsum(grad[,2]*grad[,2],CF1)/s2 + invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))

  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <- rowsum(grad[,1]*grad[,1],CF2)/s2 + invD2[1,1]
  diag(G.b2.12) <- rowsum(grad[,1]*grad[,2],CF2)/s2 + invD2[1,2]
  diag(G.b2.22) <- rowsum(grad[,2]*grad[,2],CF2)/s2 + invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))
```

```
# compute part G related to both b1 & b2
G.b12.11 <- matrix(rowsum(grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T,
                   ncol = n2)
G.b12.12 <- matrix(rowsum(grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T,
                   ncol = n2)
G.b12.22 <- matrix(rowsum(grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T,
                   ncol = n2)
G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))
G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

G <- G.b # finalize G
log.detG <- determinant(G, logarithm = TRUE)$modulus

value <- 0.5*N*log(2*pi*s2) + 0.5*n1*log(det(D1)) +
         0.5*n2*log(det(D2)) + 0.5*log.detG + g

# calculate 1st derivative w.r.t. fixed & random effects
invG <- solve(G)
invG.b1 <- invG[1:(2*n1),1:(2*n1)]
invG.b2 <- invG[(2*n1+1):(2*n1+2*n2),(2*n1+1):(2*n1+2*n2)]
invG.b12 <- invG[1:(2*n1),(2*n1+1):(2*n1+2*n2)]
a.gr <- colSums((y-der)*(-grad))/s2
b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
b1.gr <- b1.gr + b1 %*% invD1
b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
b2.gr <- b2.gr + b2 %*% invD2
for(jj in 1:nP){
  #### 1st derivative w.r.t. a ####
  # compute dG.b1 related to a[jj]
  dG.b1.11 <- dG.b1.12 <- dG.b1.22 <- diag(integer(n1))
  diag(dG.b1.11) <-
      rowsum(hess[,jj,1]*grad[,1]+hess[,jj,1]*grad[,1],CF1)/s2
  diag(dG.b1.12) <-
      rowsum(hess[,jj,1]*grad[,2]+hess[,jj,2]*grad[,1],CF1)/s2
  diag(dG.b1.22) <-
      rowsum(hess[,jj,2]*grad[,2]+hess[,jj,2]*grad[,2],CF1)/s2
  dG.b1 <- 0.5*rbind(cbind(dG.b1.11,dG.b1.12),
                     cbind(dG.b1.12,dG.b1.22))
  invG.dG.b1 <- invG.b1*dG.b1

  # compute dG.b2 related to a[jj]
  dG.b2.11 <- dG.b2.12 <- dG.b2.22 <- diag(integer(n2))
  diag(dG.b2.11) <-
      rowsum(hess[,jj,1]*grad[,1]+hess[,jj,1]*grad[,1],CF2)/s2
  diag(dG.b2.12) <-
      rowsum(hess[,jj,1]*grad[,2]+hess[,jj,2]*grad[,1],CF2)/s2
  diag(dG.b2.22) <-
      rowsum(hess[,jj,2]*grad[,2]+hess[,jj,2]*grad[,2],CF2)/s2
  dG.b2 <- 0.5*rbind(cbind(dG.b2.11,dG.b2.12),
                     cbind(dG.b2.12,dG.b2.22))
  invG.dG.b2 <- invG.b2*dG.b2

  # compute dG.b12 related to a[jj]
  dG.b12.11 <-
      matrix(rowsum(hess[,jj,1]*grad[,1]+hess[,jj,1]*grad[,1],
                    CF1:CF2)/s2, byrow=T, ncol = n2)
```

```
    dG.b12.12 <-
        matrix(rowsum(hess[,jj,1]*grad[,2]+hess[,jj,2]*grad[,1],
                      CF1:CF2)/s2, byrow=T, ncol = n2)
    dG.b12.22 <-
        matrix(rowsum(hess[,jj,2]*grad[,2]+hess[,jj,2]*grad[,2],
                      CF1:CF2)/s2, byrow=T, ncol = n2)
    dG.b12 <- 0.5*rbind(cbind(dG.b12.11,dG.b12.12),
                        cbind(dG.b12.12,dG.b12.22))
    invG.dG.b12 <- invG.b12*dG.b12
    a.gr[jj] <- a.gr[jj] + sum(invG.dG.b1, invG.dG.b2, 2*invG.dG.b12)

    if(jj <= 2){
      #### 1st derivative w.r.t. b1 ####
      # prepare dG.b2 related to a[jj]
      M.b2.11 <-
          rowsum(hess[,jj,1]*grad[,1]+hess[,jj,1]*grad[,1],CF1:CF2)/s2
      M.b2.12 <-
          rowsum(hess[,jj,1]*grad[,2]+hess[,jj,2]*grad[,1],CF1:CF2)/s2
      M.b2.22 <-
          rowsum(hess[,jj,2]*grad[,2]+hess[,jj,2]*grad[,2],CF1:CF2)/s2
      b1.gr[,jj] <- (b1.gr[,jj] +
          rowSums(matrix(rowSums(invG.dG.b1),ncol=2)) +
              2*rowSums(matrix(rowSums(invG.dG.b12),ncol=2)))
      b1.gr[,jj] <- (b1.gr[,jj] +
          0.5*rowSums(matrix(diag(invG.b2[1:n2,1:n2])*M.b2.11,
              byrow=T, ncol=n2)) +
          2*0.5*rowSums(matrix(diag(invG.b2[1:n2,
              (n2+1):(2*n2)])*M.b2.12,byrow=T,ncol=n2)) +
          0.5*rowSums(matrix(diag(invG.b2[(n2+1):(2*n2),
              (n2+1):(2*n2)])*M.b2.22,byrow=T,ncol=n2)))

      #### 1st derivative w.r.t. b2 ####
      # prepare dG.b1 related to a[jj]
      M.b1.11 <- rowsum(hess[,jj,1]*grad[,1] +
                  hess[,jj,1]*grad[,1],CF2:CF1)/s2
      M.b1.12 <- rowsum(hess[,jj,1]*grad[,2] +
                  hess[,jj,2]*grad[,1],CF2:CF1)/s2
      M.b1.22 <- rowsum(hess[,jj,2]*grad[,2] +
                  hess[,jj,2]*grad[,2],CF2:CF1)/s2
      b2.gr[,jj] <- (b2.gr[,jj] +
          rowSums(matrix(rowSums(invG.dG.b2),ncol=2)) +
          2*rowSums(matrix(colSums(invG.dG.b12),ncol=2)))
      b2.gr[,jj] <- (b2.gr[,jj] +
          0.5*rowSums(matrix(diag(invG.b1[1:n1,1:n1])*M.b1.11,
              byrow=T,ncol=n1)) +
          2*0.5*rowSums(matrix(diag(invG.b1[1:n1,
              (n1+1):(2*n1)])*M.b1.12,byrow=T,ncol=n1)) +
          0.5*rowSums(matrix(diag(invG.b1[(n1+1):(2*n1),
              (n1+1):(2*n1)])*M.b1.22,byrow=T,ncol=n1)))
    }
  }
  attr(value, "gradient") <- c(a.gr, b1.gr, b2.gr)
  return(value)
}
```

```
##################
# START OF D.obj #
##################

D.obj <- function(D1,D2,s2,ab) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

g.obj <- function(ab) {
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b.gr <- c(b1.gr, b2.gr)
  attr(g, "gradient") <- c(a.gr, b.gr)

  g
}

  # Update fixed & random effects
  g.optim <- nlm(g.obj, ab, iterlim = 500, print.level=0)
  g.optim
  ab <- g.optim$estimate
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)

  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  hess <- attr(der, "hessian") # 2nd derivative

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2
```

```
  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <- rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1)/s2 + invD1[1,1]
  diag(G.b1.12) <- rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1)/s2 + invD1[1,2]
  diag(G.b1.22) <- rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1)/s2 + invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))

  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <- rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF2)/s2 + invD2[1,1]
  diag(G.b2.12) <- rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF2)/s2 + invD2[1,2]
  diag(G.b2.22) <- rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF2)/s2 + invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <- rowsum((y-der)*(-hess[,1:3,1])+grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <- rowsum((y-der)*(-hess[,1:3,2])+grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <- rowsum((y-der)*(-hess[,1:3,1])+grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <- rowsum((y-der)*(-hess[,1:3,2])+grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)
  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  G <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  monitor <- (0.5*N*log(2*pi*s2) - 0.5*nP*log(2*pi)  +
              0.5*n1*log(det(D1)) + 0.5*n2*log(det(D2)) +
              0.5*determinant(G, logarithm = TRUE)$modulus +
              g.optim$minimum)

  list(val=g.optim$minimum, monitor=monitor, G=G, invG=solve(G),
       detG=determinant(G, logarithm = TRUE)$modulus, ab=ab)
}
```

```
###################
# START OF N1.obj #
###################

N1.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  d.sse.b <- (y - der)*(-grad[,1:2]) # 1st derivative of sse w.r.t. b
  hess <- attr(der, "hessian") # 2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sum(b1[,1]^2)/n1

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b1.gr[,1] <- b1.gr[,1] - 2*s*b1[,1]/n1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF1)/s2+invD1[1,1]
  diag(G.b1.11) <- diag(G.b1.11) - 2*s/n1
  diag(G.b1.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF1)/s2+invD1[1,2]
  diag(G.b1.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF1)/s2+invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF2)/s2+invD2[1,1]
  diag(G.b2.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF2)/s2+invD2[1,2]
  diag(G.b2.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF2)/s2+invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  err
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
###################
# START OF N2.obj #
###################

N2.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  d.sse.b <- (y - der)*(-grad[,1:2]) # 1st derivative of sse w.r.t. b
  hess <- attr(der, "hessian") # 2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sum(b1[,2]^2)/n1

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b1.gr[,2] <- b1.gr[,2] - 2*s*b1[,2]/n1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF1)/s2+invD1[1,1]
  diag(G.b1.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF1)/s2+invD1[1,2]
  diag(G.b1.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF1)/s2+invD1[2,2]
  diag(G.b1.22) <- diag(G.b1.22) - 2*s/n1
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF2)/s2+invD2[1,1]
  diag(G.b2.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF2)/s2+invD2[1,2]
  diag(G.b2.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF2)/s2+invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  err
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
####################
# START OF N12.obj #
####################

N12.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  d.sse.b <- (y - der)*(-grad[,1:2]) # 1st derivative of sse w.r.t. b
  hess <- attr(der, "hessian") # 2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sum(b1[,1]*b1[,2])/n1

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b1.gr[,1] <- b1.gr[,1] - s*b1[,2]/n1
  b1.gr[,2] <- b1.gr[,2] - s*b1[,1]/n1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF1)/s2+invD1[1,1]
  diag(G.b1.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF1)/s2+invD1[1,2]
  diag(G.b1.12) <- diag(G.b1.12) - s/n1
  diag(G.b1.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF1)/s2+invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF2)/s2+invD2[1,1]
  diag(G.b2.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF2)/s2+invD2[1,2]
  diag(G.b2.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF2)/s2+invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
###################
# START OF M1.obj #
###################

M1.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  d.sse.b <- (y - der)*(-grad[,1:2]) # 1st derivative of sse w.r.t. b
  hess <- attr(der, "hessian") # 2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sum(b2[,1]^2)/n2

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b2.gr[,1] <- b2.gr[,1] - 2*s*b2[,1]/n2
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <-
    rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF1)/s2+invD1[1,1]
  diag(G.b1.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF1)/s2+invD1[1,2]
  diag(G.b1.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF1)/s2+invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF2)/s2+invD2[1,1]
  diag(G.b2.11) <- diag(G.b2.11) - 2*s/n2
  diag(G.b2.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF2)/s2+invD2[1,2]
  diag(G.b2.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF2)/s2+invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
###################
# START OF M2.obj #
###################

M2.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  d.sse.b <- (y - der)*(-grad[,1:2]) # 1st derivative of sse w.r.t. b
  hess <- attr(der, "hessian") # 2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sum(b2[,2]^2)/n2

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b2.gr[,2] <- b2.gr[,2] - 2*s*b2[,2]/n2
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF1)/s2+invD1[1,1]
  diag(G.b1.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF1)/s2+invD1[1,2]
  diag(G.b1.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF1)/s2+invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF2)/s2+invD2[1,1]
  diag(G.b2.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF2)/s2+invD2[1,2]
  diag(G.b2.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF2)/s2+invD2[2,2]
  diag(G.b2.22) <- diag(G.b2.22) - 2*s/n2
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
####################
# START OF M12.obj #
####################

M12.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  d.sse.b <- (y - der)*(-grad[,1:2]) # 1st derivative of sse w.r.t. b
  hess <- attr(der, "hessian") # 2nd derivative

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sum(b2[,1]*b2[,2])/n2

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums((y - der)*(-grad))/s2

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum((y - der)*(-grad[,1:2]),CF1)/s2
  b1.gr <- b1.gr + b1 %*% invD1
  b2.gr <- rowsum((y - der)*(-grad[,1:2]),CF2)/s2
  b2.gr <- b2.gr + b2 %*% invD2
  b2.gr[,1] <- b2.gr[,1] - s*b2[,2]/n2
  b2.gr[,2] <- b2.gr[,2] - s*b2[,1]/n2
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad))/s2

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF1)/s2+invD1[1,1]
  diag(G.b1.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF1)/s2+invD1[1,2]
  diag(G.b1.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF1)/s2+invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <-
      rowsum((y-der)*(-hess[,1,1])+grad[,1]*grad[,1],CF2)/s2+invD2[1,1]
  diag(G.b2.12) <-
      rowsum((y-der)*(-hess[,1,2])+grad[,1]*grad[,2],CF2)/s2+invD2[1,2]
  diag(G.b2.12) <- diag(G.b2.12) - s/n2
  diag(G.b2.22) <-
      rowsum((y-der)*(-hess[,2,2])+grad[,2]*grad[,2],CF2)/s2+invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2)/s2, byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF1)/s2
  G.ab1.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF1)/s2
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <-
      rowsum((y-der)*(-hess[,1:3,1]) + grad[,1:3]*grad[,1],CF2)/s2
  G.ab2.2 <-
      rowsum((y-der)*(-hess[,1:3,2]) + grad[,1:3]*grad[,2],CF2)/s2
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
###################
# START OF s2.obj #
###################

s2.obj <- function(D1,D2,s2,ab,s) {
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  for (jj in 1:20){
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)]
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  phi1 <- a[1] + b1[CF1,1] + b2[CF2,1]
  phi2 <- a[2] + b1[CF1,2] + b2[CF2,2]
  phi3 <- a[3]
  der <- eval(der.exp) # f value
  grad <- attr(der, "gradient") # 1st derivative
  hess <- attr(der, "hessian") # 2nd derivative
  sse <- as.numeric(crossprod(y - der))
  d.sse <- (y - der)*(-grad)

  # calculate g
  g <- crossprod(y-der)/(2*s2)
  g <- g + sum(b1 %*% invD1 * b1)/2 + sum(b2 %*% invD2 * b2)/2
  g <- g - s*sse/N

  # calculate 1st derivative of g w.r.t. a
  a.gr <- colSums(d.sse) * (1/s2 - 2*s/N)

  # calculate 1st derivative of g w.r.t. b
  b1.gr <- rowsum(d.sse[,1:2], CF1) * (1/s2 - 2*s/N)
  b1.gr <- b1.gr + crossprod(t(b1), invD1)
  b2.gr <- rowsum(d.sse[,1:2], CF2) * (1/s2 - 2*s/N)
  b2.gr <- b2.gr + crossprod(t(b2), invD2)
  b.gr <- c(b1.gr, b2.gr)
  g.gradient <- c(a.gr, b.gr)

  # compute part G related to a
  G.a <- (colSums((y-der)*(-hess)) + crossprod(grad)) * (1/s2 - 2*s/N)

  # compute part G related to b1
  G.b1.11 <- G.b1.12 <- G.b1.22 <- diag(integer(n1))
  diag(G.b1.11) <- rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1], CF1) * (1/s2 - 2*s/N) + invD1[1,1]
  diag(G.b1.12) <- rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2], CF1) * (1/s2 - 2*s/N) + invD1[1,2]
  diag(G.b1.22) <- rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2], CF1) * (1/s2 - 2*s/N) + invD1[2,2]
  G.b1 <- rbind(cbind(G.b1.11,G.b1.12), cbind(G.b1.12,G.b1.22))
```

```
  # compute part G related to b2
  G.b2.11 <- G.b2.12 <- G.b2.22 <- diag(integer(n2))
  diag(G.b2.11) <- rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF2) * (1/s2 - 2*s/N) + invD2[1,1]
  diag(G.b2.12) <- rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF2) * (1/s2 - 2*s/N) + invD2[1,2]
  diag(G.b2.22) <- rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF2) * (1/s2 - 2*s/N) + invD2[2,2]
  G.b2 <- rbind(cbind(G.b2.11,G.b2.12), cbind(G.b2.12,G.b2.22))

  # compute part G related to both b1 & b2
  G.b12.11 <- matrix(rowsum((y-der)*(-hess[,1,1]) +
      grad[,1]*grad[,1],CF1:CF2) * (1/s2 - 2*s/N), byrow=T, ncol = n2)
  G.b12.12 <- matrix(rowsum((y-der)*(-hess[,1,2]) +
      grad[,1]*grad[,2],CF1:CF2) * (1/s2 - 2*s/N), byrow=T, ncol = n2)
  G.b12.22 <- matrix(rowsum((y-der)*(-hess[,2,2]) +
      grad[,2]*grad[,2],CF1:CF2) * (1/s2 - 2*s/N), byrow=T, ncol = n2)
  G.b12 <- rbind(cbind(G.b12.11,G.b12.12), cbind(G.b12.12,G.b12.22))

  G.b <- rbind(cbind(G.b1, G.b12), cbind(t(G.b12),G.b2))

  # compute part G related to a & b1
  G.ab1.1 <- rowsum((y-der)*(-hess[,1:3,1]) +
      grad[,1:3]*grad[,1],CF1) * (1/s2 - 2*s/N)
  G.ab1.2 <- rowsum((y-der)*(-hess[,1:3,2]) +
      grad[,1:3]*grad[,2],CF1) * (1/s2 - 2*s/N)
  G.ab1 <- rbind(G.ab1.1, G.ab1.2)

  # compute part G related to a & b2
  G.ab2.1 <- rowsum((y-der)*(-hess[,1:3,1]) +
      grad[,1:3]*grad[,1],CF2) * (1/s2 - 2*s/N)
  G.ab2.2 <- rowsum((y-der)*(-hess[,1:3,2]) +
      grad[,1:3]*grad[,2],CF2) * (1/s2 - 2*s/N)
  G.ab2 <- rbind(G.ab2.1, G.ab2.2)

  G.ab <- rbind(G.ab1, G.ab2)

  # finalize G
  g.hessian <- rbind(cbind(G.a,t(G.ab)), cbind(G.ab,G.b))

  ab.new <- ab - solve(g.hessian) %*% g.gradient
  err <- max(abs(ab.new - ab) - 1.e-8*(abs(ab) + 1.e-6))
  ab <- ab.new
  if(err < 0) break
  }

  list(val=g, detG=determinant(g.hessian, logarithm = TRUE)$modulus)
}
```

```
################
# Main Program #
################


emup <- function(theta,ab) {
  D1 <- matrix(c(theta[1:2],theta[2:3]), ncol=2)
  D2 <- matrix(c(theta[4:5],theta[5:6]), ncol=2)
  s2 <- theta[7]
  result.D <- D.obj(D1,D2,s2,ab)
  ab <- result.D$ab
  a <- ab[1:nP]
  b <- ab[(nP+1):length(ab)] # random effects of CF 1
  b1 <- matrix(b[1:(2*n1)], ncol = 2)
  b2 <- matrix(b[(2*n1+1):length(b)], ncol = 2)
  s <- sqrt(.Machine$double.eps)

  N1.ps <- N1.obj(D1,D2,s2,ab,s)
  N1.ms <- N1.obj(D1,D2,s2,ab,-s)
  N1.m1 <- exp(.5*(result.D$detG - N1.ps$detG) +
      result.D$val - N1.ps$val)
  N1.m2 <- exp(.5*(result.D$detG - N1.ms$detG) +
      result.D$val - N1.ms$val)

  N2.ps <- N2.obj(D1,D2,s2,ab,s)
  N2.ms <- N2.obj(D1,D2,s2,ab,-s)
  N2.m1 <- exp(.5*(result.D$detG - N2.ps$detG) +
      result.D$val - N2.ps$val)
  N2.m2 <- exp(.5*(result.D$detG - N2.ms$detG) +
      result.D$val - N2.ms$val)

  N12.ps <- N12.obj(D1,D2,s2,ab,s)
  N12.ms <- N12.obj(D1,D2,s2,ab,-s)
  N12.m1 <- exp(.5*(result.D$detG - N12.ps$detG) +
      result.D$val - N12.ps$val)
  N12.m2 <- exp(.5*(result.D$detG - N12.ms$detG) +
      result.D$val - N12.ms$val)

  M1.ps <- M1.obj(D1,D2,s2,ab,s)
  M1.ms <- M1.obj(D1,D2,s2,ab,-s)
  M1.m1 <- exp(.5*(result.D$detG - M1.ps$detG) +
      result.D$val - M1.ps$val)
  M1.m2 <- exp(.5*(result.D$detG - M1.ms$detG) +
      result.D$val - M1.ms$val)

  M2.ps <- M2.obj(D1,D2,s2,ab,s)
  M2.ms <- M2.obj(D1,D2,s2,ab,-s)
  M2.m1 <- exp(.5*(result.D$detG - M2.ps$detG) +
      result.D$val - M2.ps$val)
  M2.m2 <- exp(.5*(result.D$detG - M2.ms$detG) +
      result.D$val - M2.ms$val)
```

```
  M12.ps <- M12.obj(D1,D2,s2,ab,s)
  M12.ms <- M12.obj(D1,D2,s2,ab,-s)
  M12.m1 <- exp(.5*(result.D$detG - M12.ps$detG) +
      result.D$val - M12.ps$val)
  M12.m2 <- exp(.5*(result.D$detG - M12.ms$detG) +
      result.D$val - M12.ms$val)

  s2.ps <- s2.obj(D1,D2,s2,ab,s)
  s2.ms <- s2.obj(D1,D2,s2,ab,-s)
  s2.m1 <- exp(.5*(result.D$detG - s2.ps$detG) +
      result.D$val - s2.ps$val)
  s2.m2 <- exp(.5*(result.D$detG - s2.ms$detG) +
      result.D$val - s2.ms$val)

  D1[1,1] <- (N1.m1 - N1.m2)/(2*s)
  D1[2,2] <- (N2.m1 - N2.m2)/(2*s)
  D1[1,2] <- (N12.m1 - N12.m2)/(2*s)
  D2[1,1] <- (M1.m1 - M1.m2)/(2*s)
  D2[2,2] <- (M2.m1 - M2.m2)/(2*s)
  D2[1,2] <- (M12.m1 - M12.m2)/(2*s)
  s2 <- (s2.m1 - s2.m2)/(2*s)

  list(theta = c(D1[1,1],D1[1,2],D1[2,2],D2[1,1],D2[1,2],D2[2,2],s2),
       ab=ab, monitor = result.D$monitor)
}


# initial values
a <- a.ini
b <- integer(2*n1+2*n2)
ab <- c(a,b)
D1 <- D1.ini
D2 <- D2.ini
s2 <- s2.ini

i <- 0
theta <- c(D1[1,1],D1[1,2],D1[2,2], D2[1,1],D2[1,2],D2[2,2], s2)
cat("\n")
while(i < 300){
  temp <- emup(theta,ab)
  thet2 <- temp$theta
  ab <- temp$ab
  i <- i + 1
  err <- max(abs(thet2 - theta) - 1.e-3*(abs(theta) + 1.e-1))
  D1.temp <- matrix(c(thet2[1:2],thet2[2:3]), ncol=2)
  D2.temp <- matrix(c(thet2[4:5],thet2[5:6]), ncol=2)
  if(min(eigen(D1.temp)$values) <= 0 | min(eigen(D2.temp)$values) <= 0
   | thet2[7] <= 0) {
    thet2[1] <- abs(thet2[1])
    thet2[2] <- 0
    thet2[3] <- abs(thet2[3])
    thet2[4] <- abs(thet2[4])
    thet2[5] <- 0
    thet2[6] <- abs(thet2[6])
    thet2[7] <- abs(thet2[7])
  }
```

```
  theta <- thet2
  cat(i,temp$monitor,err,theta,"\n")
  if(err <0 | i >= 100) break
}

A <- -diag(length(theta))
temp <- emup(theta,ab)
gg <- temp$theta - theta
monitor <- temp$monitor
ab <- temp$ab
while(i < 300){
  i <- i + 1
  deltheta <- -A %*% gg
  thet2 <- theta + deltheta
  D1.temp <- matrix(c(thet2[1:2],thet2[2:3]), ncol=2)
  D2.temp <- matrix(c(thet2[4:5],thet2[5:6]), ncol=2)
  if(min(eigen(D1.temp)$values) <= 0 | min(eigen(D2.temp)$values) <= 0
   | thet2[7] <= 0) {
    A <- -diag(length(theta))
    gg <- thetaem - theta
    deltheta <- -A %*% gg
    thet2 <- theta + deltheta
  }
  D1.temp <- matrix(c(thet2[1:2],thet2[2:3]), ncol=2)
  D2.temp <- matrix(c(thet2[4:5],thet2[5:6]), ncol=2)
  if(min(eigen(D1.temp)$values) <= 0 | min(eigen(D2.temp)$values) <= 0
   | thet2[7] <= 0) break
  err <- max(abs(thet2 - theta) - 1.e-6*(abs(theta) + 1.e-4))
  err2 <- max(abs(thet2 - theta))
  temp <- emup(thet2,ab)
  monitor <- temp$monitor
  thetaem <- temp$theta
  delgg <- thetaem - thet2 - gg
  adgg <- A %*% delgg
  A <- A + outer(c(deltheta - adgg)/sum(deltheta*adgg), c(t(A) %*%
   deltheta))
  theta <- thet2
  gg <- gg + delgg
  ab <- temp$ab
  cat(i,temp$monitor,err,theta,"\n")
  if(err < 0 | err2 < 1.e-6) break
}

if(i >= 300 | err < 0 | err2 < 1.e-6) {
  D1 <- matrix(c(theta[1:2],theta[2:3]), ncol=2)
  D2 <- matrix(c(theta[4:5],theta[5:6]), ncol=2)
  s2 <- theta[7]
  invD1 <- solve(D1)
  invD2 <- solve(D2)

  nllike.optim <- nlm(nllike.obj, ab, iterlim = 500, print.level=0)
  ab <- nllike.optim$estimate
  dat.out <- data.frame(i,err,ab[1],ab[2],ab[3],t(theta))
}
```