A Simulation Study of the Second-order Least Squares Estimators for Nonlinear Mixed Effects Models

By

He Li

A Thesis Submitted to the Faculty of Graduate Studies of
the University of Manitoba
in Partial Fulfilment of the Requirements of the Degree of
Master of Science

Department of Statistics
University of Manitoba
Winnipeg

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

A SIMULATION STUDY OF THE SECOND-ORDER LEAST SQUARES ESTIMATORS FOR NONLINEAR MIXED EFFECTS MODELS

BY

He Li

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree Of

Master of Science

Contents

List of Figures

List of Tables

Abstract

The main approach for the estimation of nonlinear mixed effects models focuses on the maximum likelihood method. Given the current computing capacity, intensive numerical integration often makes exact maximum likelihood estimation impractical. Wang (2005) proposed the second-order least squares estimators for nonlinear mixed effects models based on the first two conditional moments of the response variable given the observed predictor variables.

In this thesis, we present numerical examples demonstrating that Wang's (2005) second-order least squares estimators are computationally feasible and practical. In particular, we show how Wang's (2005) algorithm can be implemented in the statistical computing language R. Finally, we investigate the finite sample properties of the second-order least squares estimators through simulation studies.

Acknowledgements

I would like to thank Dr. Liqun Wang, my supervisor, for his many suggestions and constant support during this research. I also thank him for supporting me through his Natural Sciences and Engineering Research Council of Canada (NERSC) research grant.

I also wish to thank Dr. James C. Fu and Dr. Wendy Y. Lou who shared with me their knowledge and ideas, and provided me with many useful comments and suggestions.

The University of Manitoba Graduate Fellowship (UMGF) awarded to me was crucial to the successful completion of this thesis.

Of course, I am grateful to my parents and my girl friend for their patience, love and support.

Chapter 1

Introduction

1.1 Data and Examples

1.1.1 Example 1: Pharmacokinetics of Cefamandole

Pharmacokinetics is the study of the bodily absorption, distribution, metabolism, and excretion of drugs. The goal of pharmacokinetics modeling is to summarize the concentration-time measurements using a model that relates drug input to drug response, to relate the parameters of this model to patients' characteristics, and to provide individual dose-response predictions for use in optimizing individual doses.

The data are obtained during a pilot study to investigate the pharmacokinetics of the drug cefamandole (Davidian and Giltinan 1995). In the experiment, a dose of 15 mg/kg body weight of cefamandole is administered by ten-minute intravenous infusion to six healthy male volunteers. Plasma concentration of the drug is measured on six volunteers at 14 time points,

and the data are shown in Table 1.1 and Figure 1.1.

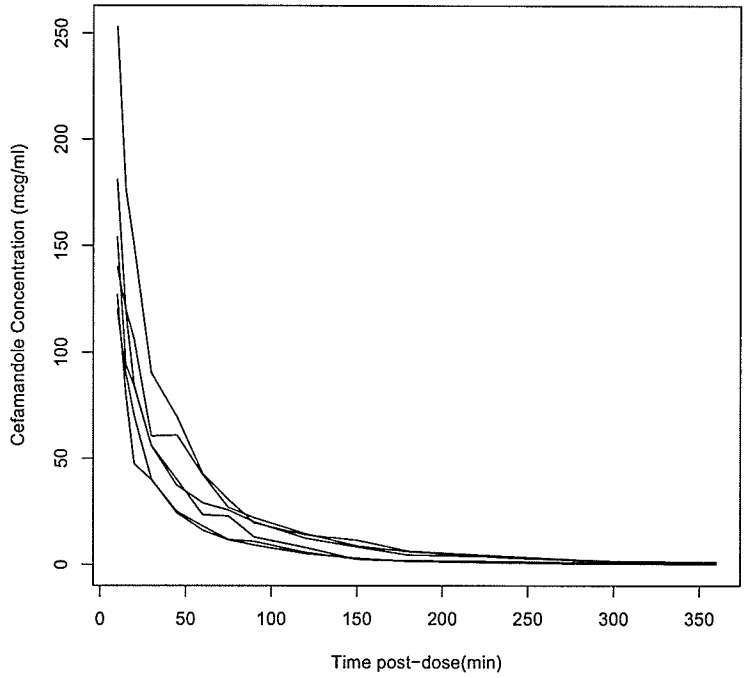

Figure 1.1: Plasma concentration-time of cefamandole versus time post injection for six subjects

We can observe that the data are collected by observing a number of subjects (units) repeatedly and responses are measured over time; all subjects' concentration profiles have a similar shape; however, peak concentration achieved, decay and elimination vary substantially.

Table 1.1: Plasma concentration-time of cefamandole versus time post injection for six subjects

	Subject					
Time	1	2	3	4	5	6
10	127.00	120.00	154.00	181.00	253.00	140.00
15	80.00	90.10	94.00	119.00	176.00	120.00
20	47.40	70.00	84.00	84.30	150.00	106.00
30	39.90	40.10	56.00	56.10	90.30	60.40
45	24.80	24.00	37.10	39.80	69.60	60.90
60	17.90	16.10	28.90	23.30	42.50	42.20
75	11.70	11.60	25.50	22.70	30.60	26.80
90	10.90	9.20	20.00	13.00	19.60	22.00
120	5.70	5.20	12.40	8.00	13.80	14.50
150	2.55	3.00	8.30	2.40	11.40	8.80
180	1.84	1.54	4.50	1.60	6.30	6.00
240	1.50	0.73	3.40	1.10	3.80	3.00
300	0.70	0.37	1.70	0.48	1.55	1.30
360	0.34	0.19	1.19	0.29	1.22	1.03

1.1.2 Example 2: Growth of Orange Trees

In a growth model studied by Draper and Smith (1981) and later by Lindstrom and Bates (1990), the data consist of seven measurements of the trunk circumference (in millimeters) on each of five orange trees, taken over a period of 1600 days. The data is reported in Table 1.2.

Table 1.2: Measurements of trunk circumference for five orange trees

	Tree				
Age (Days)	1	2	3	4	5
118	30	33	30	32	30
484	58	69	51	62	49
664	87	111	75	112	81
1004	115	156	108	167	125
1231	120	172	115	179	142
1372	142	203	139	209	174
1582	145	203	140	214	177

From Table 1.2 and Figure 1.2, we can see that the data are collected by observing a number of trees (units) repeatedly and responses are measured over time; all growth curves have a similar shape; however, the growth rate of each curve is significantly different.

In both example 1 and 2, data are collected repeatedly and responses are measured over different units. This type of data is called repeated mea-

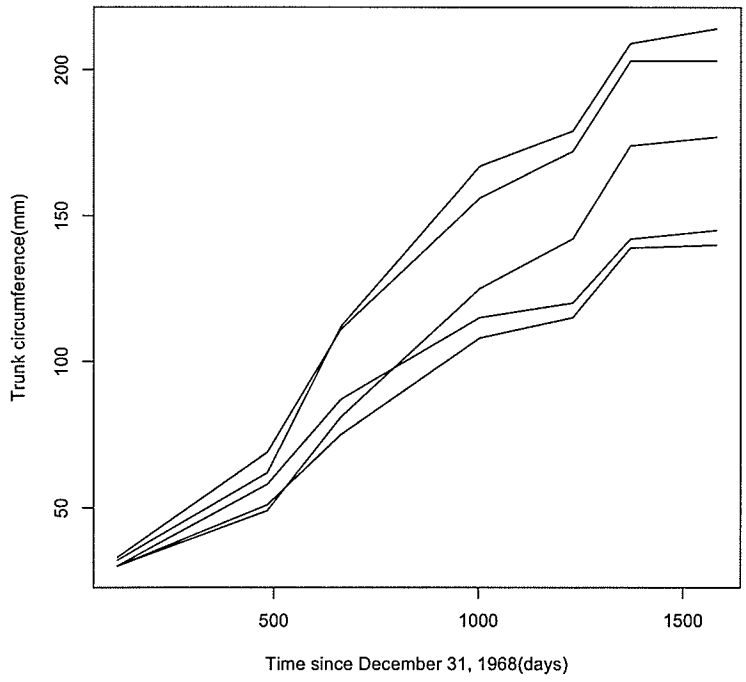

Figure 1.2: Measurements of trunk circumference for five orange trees

surement data. In general, by "repeated measurement data", we mean data are generated by observing a number of units repeatedly and responses are measured under different experiment conditions. A common type of repeated measurement data is longitudinal data. Many longitudinal studies are designed to investigate changes over time in a characteristic which is measured repeatedly for each study subject (Laird and Ware 1982), such as the cefamandole data in example 1 and the orange tree data in example 2. However, the models and methods we will discuss are more broadly applicable to any kind of repeated measurement data which could be measured over some other set of conditions, such as at different positions in space, or across different concentrations and dosages.

1.2 Mixed Effects Models

By observing the cefamandole and orange tree data, we can find they share some similar characteristics. All repeated response measurements are taken at different time for different groups. There is a nonlinear dependence of the response on parameters of interest. All units have a similar shaped profile but with noticeable intra-group and within-group variabilities .

For the pharmacokinetics study in example 1, the purpose is to study what the drug does to the body and determine the appropriate regimen of dosages for different individuals. As shown in Figure 1.1, although the concentration profiles have a similar shape for all individuals, peak concentra-

tion achieved, rise and decay vary substantially. These substantial differences among individuals are attributable to the intra-individual variations which is associated with an individual's demographic, physiological and behavioral characteristics. From investigators' point, the use of population pharmacokinetics in the drug development process should help identify differences in drug safety and efficacy among subgroups. For the growth study in example 2, biologists are interested in the description of different growth patterns and in trying to understand the underlying mechanisms. From Figure 1.2, all trees have similar growth curves, but the growth rate is different among trees. It is critical for biologists to understand the variations to determine how growth responds to different treatments or covariate information.

Results of the pilot pharmacokinetics and growth studies are to be used as a basis for subsequent investigation in a more heterogeneous group population. The focus of these studies is not on population mean, but on the group parameters in the population. Due to the similar profiles and substantial variation among groups, each group may have the same model but with different parameters. The use of simple nonlinear regression analysis, which can only capture the profile of a homogenous group, is not informative enough for these types of studies. Therefore, some hierarchical statistical models and methods are needed to acknowledge these unique features in the repeated measurement data.

Mixed effects models, in which the regression coefficients are allowed to

vary across units, are commonly used to incorporate both variations within and between units. They include a mixture of fixed effects, which are parameters associated with the entire population, and random effects which are associated with individual experimental units. They can not only describe the trend of data over time while taking account of the correlation that exists between successive measurements, but also describe the different variation for each unit over time. They provide a powerful technique for the analysis of repeated measurement data that arise in many applied fields.

The linear mixed effects models are easy to handle, and well applied for evaluating the performance of products, for determining sampling designs, and quality-control procedures, and particularly for analyzing longitudinal data (Lee and Xu 2004). However, the data sets, in many studies , such as growth studies, clinical research or pharmacokinetic and pharmacodynamic studies (Davidian and Giltinan 1995; Vonesh and Chinchilli 1997; Lindsey 1999), are nonlinear in nature with respect to a given response regression function. Therefore, nonlinear mixed effects models are required to fit this type of data. Many different nonlinear mixed effects models have been proposed in recent years (Sheiner and Beal 1980; Mallet, Mentre, Steimer and Lokiek 1988; Lindstrom and Bates 1990; Davidian and Gallant 1992; Vonesh and Carter 1992).

1.3 Statistical Inference

Nonlinear mixed effects models have been receiving increased attention in recent years, because many quantitative relationships are nonlinear inherently and can not be simply approximated by linear ones. Although they are more realistic in many fields, the estimation always raises many theoretical and computational challenges due to the fact that these models are typically nonlinear with respect to the random effects.

1.3.1 Maximum Likelihood Estimation Approach

The main estimation approach for nonlinear mixed effects models is maximum likelihood approach. However, it requires complete specifications of distributions for all random variables. To facilitate the implementation of the approach, normal distributions are always assumed. The major challenge of the maximum likelihood method is that numerical computation is difficult or intractable because the likelihood function involves multiple integrals and dose not have a closed-form expression. Given the current computing capacity, intensive numerical integration often makes exact maximum likelihood estimation impractical. Therefore, various approaches are proposed to approximate the likelihood function based on the normality assumptions to alleviate the computational burden and instability associated with complex numerical integration.

Linearization of the likelihood function for nonlinear mixed effects mod-

els is by far the most popular technique. It was first proposed by Sheiner and Beal (1980). They linearized the nonlinear response function of the model with a first order Taylor series expansion, and then fitted the resulting linearized model by the maximum likelihood approach. Laird and Ware (1982) viewed the data of nonlinear mixed effects models that could be modeled by an expectation function that was linear in its parameters. Lindstrom and Bates (1990) proposed a more accurate approximation to the nonlinear response function of the model by expanding the nonlinear response function of the model about the current estimates of the fixed effects and the random effects. They used Newton-Raphson algorithm to carry out the maximum likelihood fit of the linearized model. We will further introduce Lindstrom and Bates (1990) linearization algorithm in the next chapter. Other linerization methods include, Vonesh and Carter (1992), Liang and Zeger (1986), Goldstein (1991), and Longford (1994)

Some other approximate maximum approaches have also been intensively studied in recent years. They include the EM algorithm (Walker 1996 and Yang 2001), Laplacian approximation (Wolfinger 1993 and Vonesh 1996), "exact" likelihood algorithm (Pinherio and Bates 1995, Davidian and Gallant 1993 and Dempster 1997), and spline approximation (Ge, 2003). Pinherio and Bates (1995) gave a comprehensive review on most of the parametric approaches.

All these likelihood approaches are based on the approximated likeli-

hood function and rely on the normality assumptions of random effects.

1.3.2 Second-order Least Squares Estimation Approach

Wang (2003, 2004) has shown that the nonlinear Berkson measurement error models are generally identifiable using the first two conditional moments of the response variable given the observed predictor variables. Wang (2004) proposes a minimum distance-type estimator and a simulation-based estimator based on the first two conditional moments of the response variable. By demonstrating the same identifiability property for nonlinear mixed effects models, Wang (2005) proposes the second-order least squares (SLS) approach by extending Wang's (2004) method to the estimation of nonlinear mixed effects models. The second-order least squares estimators (SLSE) are straightforward to compute, if the closed forms of the two conditional moments are available; otherwise, Wang (2005) proposes a simulation-based estimator (SBE) by approximating the two conditional moments using Monte Carlo techniques. Wang (2005) has shown that both SLSE and SBE are consistent and asymptotically normally distributed under fairly general regularity conditions.

For the second-order least squares approach, it is computational feasible and does not depend on the normality assumptions for random effects.

1.3.3 Other Approaches

A semiparametric modeling approach was proposed by Gallant and Nychka (1987), and further developed by Davidian and Gallant (1993). Some nonparametric method were proposed by Mallet (1992), Mentre and Mallet (1994), and Lai (2003). Steimer et al. (1984) proposed an iterative two-stage method for estimating nonlinear random effects model. Lu and Meeker (1993) proposed a two-stage estimation method which was similar to the Steimer et al. (1984) method. Mentre and Gomeni (1995) proposed a two-step iterative algorithm which could be viewed as an approximation to the EM algorithm used by Walker (1996). Other approaches involve evaluation of the integrals via numerical integration or using Markov chain Monte Carlo simulation techniques, including Davidian and Gallant (1993), Pinheiro and Bates (1995b), and Concordet and Nunez (2001). Several Bayesian approaches were proposed by Berkey (1982), Racine-Poon (1985) and Gilks et. al. (1996).

1.4 Scope of the Thesis

We conduct several simulation studies in this thesis to show how the numerical computation of second-order least squares approach can be implemented in the statistical computing language R. Based on the simulation results, we investigate the finite sample properties of the second-order least squares estimators. We also examine how the SLSE estimation approach performs

for the non-normal distributions of random effects. Finally, we apply this method to two real data sets.

The thesis is organized as follows. In chapter 2, we introduce the Lindstrom and Bates' (1990) linearized likelihood approach and Wang's (2005) second-order least squares approach. Chapter 3, we briefly review some nonlinear numerical optimization methods and a concise comparison is given. In chapter 4, simulation studies are conducted to investigate how well second-order least squares method performs for finite samples. In chapter 5, we perform several simulation studies based on the non-normal random effects. Two real data applications are given in chapter 6. In chapter 7, we give an overall summary about our simulation studies and discuss some further extensions of the second-order least squares estimation method.

Chapter 2

Estimation of Nonlinear Mixed Effects Models

In the first section of this chapter, we review one linear approximation method proposed by Lindstrom and Bates (1990) to estimate nonlinear mixed effects models. The reason we choose to review this particular method is due to its popularity and availability of a variety of statistical softwares to implement this method. In the second section, we introduce the second-order least squares estimation method proposed by Wang (2005).

2.1 Linear Approximation to the Likelihood Function

Lindstrom and Bates (1990) define nonlinear mixed effects model for the jth observation on the ith individual as

$$y_{ij} = f(\phi_i, x_{ij}) + e_{ij}, \;\; i = 1, \ldots, M, \;\; j = 1, \ldots, n_i,$$

where y_{ij} is the jth response on the ith individual, x_{ij} is the predictor vector, ϕ_i is a parameter vector, and e_{ij} is a normally distributed error term. The parameter vector varies from individual to individual , so ϕ_i can be written as

$$\phi_i = A_i\beta + B_i b_i, \;\; b_i \sim N(0, \sigma^2 D), \tag{2.1}$$

where β is a $p \times 1$ vector of fixed effects, b_i is a $q \times 1$ vector of random effects associated only with individual i, matrices A_i and B_i are $r \times p$ and $r \times q$ design matrices for the fixed and random effects, respectively, and $\sigma^2 D$ is the variance-covariance matrix of the random effects.

The maximum likelihood estimation is based on the marginal density of y

$$p(y|\beta, D, \sigma^2) = \int p(y|b, \beta, D, \sigma^2)p(b)db \tag{2.2}$$

Because the model function $f(\phi_i, x_{ij})$ is nonlinear with respect to random effects, the integration (2.2) generally does not have a closed form; therefore, the numerical optimization of the likelihood function is burdensome.

Lindstrom and Bates (1990) take a first-order Taylor expansion of the model function f around the conditional modes of the random effects to approximate the log-likelihood function. The estimation algorithm described by Lindstrom and Bates (1990) proceeds in two alternating steps, a penalized nonlinear least squares (PNLS) step, which updates the estimates of fixed effects and conditional mode of random effects, and a linear mixed effects

(LME) step, which updates the variance-covariance matrix of the random effects. The algorithm alternates between PNLS and LME steps until a certain convergence criterion is met.

This linearization approach is numerically simple and efficient. It is implemented in popular software packages such as nlme() in S-PLUS and R, and PROC NLMIXED in SAS.

2.2 Second-order Least Squares Estimation

Following Wang (2005), we define the general nonlinear mixed effects model for the jth observation on the ith individual as

$$y_{ij} = g(x_{ij}, \xi_i, \gamma) + \epsilon_{ij}, \;\; i = 1, 2, \ldots, N, \;\; j = 1, 2, \ldots, n_i; \tag{2.3}$$

where $y_{ij} \in \mathbb{R}$ is the response variable, $x_{ij} \in \mathbb{R}^l$ is the predictor variable, and $\xi_i \in \mathbb{R}^m$ are unknown parameters, and ϵ_{ij} is the random error. Further, assume that

$$\xi_i = Z_i\theta + \delta_i, \tag{2.4}$$

where $Z_i \in \mathbb{R}^{m \times q}$ is a matrix of explanatory variables, $\theta \in \mathbb{R}^q$ is the vector of fixed effects and $\delta_i \in \mathbb{R}^m$ is the vector of random effects, which is independent of Z_i and $X_i = (x_{i1}, x_{i2}, \ldots, x_{in_i})'$. δ_i is independent and identically distributed with density $f_\delta(u, \phi)$, where $\phi \in \mathbb{R}^r$ is an unknown parameter.

Wang (2005) has shown that the nonlinear mixed effects models are generally identifiable using the first two conditional moments of the response

variable given the observed predictor variables.

2.2.1 The Second-order Least Squares Estimators

Under the model assumptions, the conditional mean of y_{ij} is given by

$$E(y_{ij}|X_i, Z_i) = \int g(x_{ij}, u, \gamma) f_\delta(u - Z_i\theta; \phi) du \tag{2.5}$$

and the second moments of y_{ij} given X_i, Z_i are

$$E(y_{ij}y_{ik}|X_i, Z_i) = \int g(x_{ij}, u, \gamma) g(x_{ik}, u, \gamma) f_\delta(u - Z_i\theta; \phi) du + \sigma_{ijk} \tag{2.6}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise. Let $\psi=(\theta', \gamma', \phi', \sigma_\epsilon^2)'$, $\mu_{ij}(\psi) = E(y_{ij}|X_i, Z_i)$ and $\nu_{ijk}(\psi) = E(y_{ij}y_{ik}|X_i, Z_i)$. The SLSE for ψ are defined as the measurable function satisfying

$$Q_N(\hat{\psi}) = \min_{\psi \in \Psi} Q_N(\psi) \tag{2.7}$$

where Ψ is the parameter space,

$$Q_N(\psi) = \sum_{i=1}^{N} \rho_i'(\psi)\rho_i(\psi) \tag{2.8}$$

and

$$\rho_i(\psi) = (y_{ij} - \mu_{ij}(\psi),\ 1 \le j \le n_i,\ y_{ij}y_{ik} - \nu_{ijk}(\psi),\ 1 \le j \le k \le n_i)'$$

2.2.2 The Simulation-based Estimators

If explicit forms of the integrals in (2.5) and (2.6) can not be obtained, a simulation-based approach for estimation in which the integrals are simulated

by Monte Carlo methods such as importance sampling was proposed by Wang (2005).

The simulation-based estimator is constructed by choosing a known density function $h(u)$ and generating an $i.i.d$ random sample $u_{is},\ i = 1, 2, \ldots, N$, $s = 1, 2, \ldots, 2S$. Then $\mu_{ij}(\psi)$ and $\nu_{ij}(\psi)$ are approximated by the corresponding Monte Carlo simulators

$$\mu_{ij,1}(\psi) = \frac{1}{S}\sum_{s=1}^{S} \frac{g(x_{ij}, u_{is}, \gamma) f_\delta(u_{is} - Z_i\theta; \phi)}{h(u_{is})},$$

$$\mu_{ij,2}(\psi) = \frac{1}{S}\sum_{s=S+1}^{2S} \frac{g(x_{ij}, u_{is}, \gamma) f_\delta(u_{is} - Z_i\theta; \phi)}{h(u_{is})}$$

and

$$\nu_{ijk,1}(\psi) = \frac{1}{S}\sum_{s=1}^{S} \frac{g(x_{ij}, u_{is}, \gamma) g(x_{ik}, u_{is}, \gamma) f_\delta(u_{is} - Z_i\theta; \phi)}{h(u_{is})} + \sigma_{ijk},$$

$$\nu_{ijk,2}(\psi) = \frac{1}{S}\sum_{s=S+1}^{2S} \frac{g(x_{ij}, u_{is}, \gamma) g(x_{ik}, u_{is}, \gamma) f_\delta(u_{is} - Z_i\theta; \phi)}{h(u_{is})} + \sigma_{ijk}.$$

The simulation-based estimator (SBE) for ψ is defined by

$$Q_{N,S}(\hat{\psi}) = \min_{\psi \in \Psi} Q_{N,S}(\psi) \tag{2.9}$$

where

$$Q_{N,S}(\psi) = \sum_{i=1}^{N} \rho'_{i,1}(\psi)\rho_{i,2}(\psi) \tag{2.10}$$

and

$$\rho_{i,1}(\psi) = (y_{ij} - \mu_{ij,1}(\psi),\ 1 \le j \le n_i,\ y_{ij}y_{ik} - \nu_{ijk,1}(\psi),\ 1 \le j \le k \le n_i)',$$

$$\rho_{i,2}(\psi) = (y_{ij} - \mu_{ij,2}(\psi),\ 1 \le j \le n_i,\ y_{ij}y_{ik} - \nu_{ijk,2}(\psi),\ 1 \le j \le k \le n_i)'.$$

It has been showed by Wang (2005) that $Q_{N,S}(\psi)$ is an unbiased estimator for $Q_N(\psi)$.

In practice, the choice of $h(u)$ will affect the finite sample variances of the Monte Carlo estimators such as $\mu_{ij,1}(\psi)$. Theoretically, the best choice of $h(u)$ is proportional to the absolute value of the integrand, which is $g(x_{ij}, u, \gamma) f_\delta(u - Z_i\theta; \phi)$ for $\mu_{ij,1}(\psi)$. Practically, however, a density close or being proportional to the integrand is a good choice (Wang 2004).

In the minimization of our objective function $Q_N(\psi)$ or $Q_{N,S}(\psi)$, numerical optimization methods are required because the moment functions are usually nonlinear with respect to the parameters of interest. In the next chapter, we introduce several general numerical optimization methods.

Chapter 3

Numerical Optimization Methods

A general optimization problem is to find the value of a vector $\theta \in \Theta$ that maximize or minimize a given function $Q_n(\theta)$. The function $Q_n(\theta)$ is called the objective function which depends on given observations $x_1, x_2, \cdots, x_n$, and Θ is the domain of allowable values for the vector θ. In statistics, the nonlinear least squares estimation method and maximum likelihood estimation method are two typical optimization problems.

3.1 Nonlinear Least Squares Estimation

A general nonlinear regression model can be written as:

$$y_i = f(x_i, \theta) + \epsilon_i, \quad i = 1, 2, \ldots, n \tag{3.1}$$

where θ is a $p \times 1$ vector of unknown parameter, f is a nonlinear function with respect to the parameter θ, x_i is a $k \times 1$ vector of explanatory variables

and ϵ_i is a random error term.

The nonlinear least squares estimator for θ can be obtained by minimizing the objective function $Q_n(\theta)$, where

$$Q_n(\theta) = \sum_{i=1}^{n} [y_i - f(x_i, \theta)]^2 \tag{3.2}$$

Because f is nonlinear with respect to θ, in general, it is not possible to solve explicitly for the nonlinear least square estimates by minimizing $Q_n(\theta)$ in (3.2). Therefore, numerical optimization methods must be implemented to obtain the solution.

3.2 Maximum Likelihood Estimation

The method of maximum likelihood estimation is a general method of finding estimators. Suppose y is a random variable with probability distribution $p(y;\theta)$, where θ is a $p \times 1$ vector of unknown parameters. Let $y_1, y_2, \cdots, y_n$ be a random sample of n independent observations, the likelihood function $L(\theta; y)$ can be written as

$$L(\theta; y) = \prod_{i=1}^{n} p(y_i; \theta) \tag{3.3}$$

The maximum likelihood estimator of θ is the value of θ that maximizes the likelihood function $L(\theta; y)$. Therefore, in our previous notation, $L(\theta; y)$ is our objective function $Q_n(\theta)$. If the probability distribution of y_i is nonlinear with respect to the unknown parameter θ, it is usually troublesome to find

the closed form of the maximum likelihood estimator. Therefore, numerical optimization methods are required to obtain the numerical solution.

3.3 Numerical Optimization

As we have shown in section 3.1 and 3.2, for the estimation of nonlinear models, numerical optimization methods are needed to obtain the estimates. In this section, we review several common optimization methods. For the sake of simplicity, we use the nonlinear least square estimation introduced in section 3.1 as an example to illustrate these methods.

3.3.1 Gauss-Newton Method

A popular method used in computer algorithms for the numerical optimization is the Gauss-Newton iteration method. The method is to use a linear approximation to the function $f(x_i, \theta)$ to iteratively improve an initial guess $\hat{\theta}_0$ for θ and keep improving the value of θ until there is no change. That is, we expand the function $f(x_i, \theta)$ in a first order Taylor series about $\hat{\theta}_0 = [\hat{\theta}_{10}, \hat{\theta}_{20}, \cdots, \hat{\theta}_{p0}]'$ as

$$f(x_i, \theta) \approx f(x_i, \hat{\theta}_0) + \sum_{j=1}^{p} \left[\frac{\partial f(x_i, \theta_j)}{\partial \theta_j} \right]_{\theta_j = \hat{\theta}_{j0}} (\theta_j - \hat{\theta}_{j0}) \tag{3.4}$$

If we define

$$f_i^0 = f(x_i, \hat{\theta}_0), \quad \phi_j^0 = \theta_j - \hat{\theta}_{j0}, \quad Z_{ij}^0 = \left[\frac{\partial f(x_i, \theta_j)}{\partial \theta_j} \right]_{\theta = \hat{\theta}_{j0}}$$

then the nonlinear regression model (3.1) can be approximately written as

$$y_i - f_i^0 = \sum_{j=1}^{p} \phi_j^0 Z_{ij}^0 + \epsilon_i, \quad i = 1, 2, \ldots, n \tag{3.5}$$

or further as

$$y_0 = Z_0 \phi_0 + \epsilon \tag{3.6}$$

where $y_0 = y - f_0$, $f_0 = [f_1^0, f_2^0, \ldots, f_n^0]'$, $Z_0 = [Z_{i1}^0, \ Z_{i2}^0, \ \cdots, \ Z_{ip}^0]$ and $\phi_0 = [\phi_1^0, \ \phi_2^0, , \ \cdots, \ \phi_p^0]'$. Therefore, using the least square method, ϕ_0 is computed as

$$\hat{\phi}_0 = (Z_0' Z_0)^{-1} Z_0' y_0 = (Z_0' Z_0)^{-1} Z_0' (y - f_0). \tag{3.7}$$

Now since $\phi_0 = \theta - \hat{\theta}_0$, $\hat{\theta}_1$ could be defined as,

$$\hat{\theta}_1 = \hat{\phi}_0 + \hat{\theta}_0 \tag{3.8}$$

which can be viewed as the updated value of θ. The $\hat{\theta}_1$ is substituted for $\hat{\theta}_0$ in equation (3.4). This procedure is repeated to obtain the next updated value.

In general, at the kth iteration

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \hat{\phi}_k = \hat{\theta}_k + (Z_k' Z_k)^{-1} Z_k' (y - f_k) \tag{3.9}$$

where

$$Z_k = [Z_{i1}^k, \ Z_{i2}^k, \ \cdots, \ Z_{ip}^k], \ f_k = [f_1^k, f_2^k, \ldots, f_n^k]', \ \hat{\theta}_k = [\hat{\theta}_{1k}, \hat{\theta}_{2k}, \ldots, \hat{\theta}_{pk}]'$$

The iteration process continues until convergence, that is, until

$$\left| \hat{\theta}_{j,k+1} - \hat{\theta}_{jk} \right| < \delta, \quad j = 1, 2, \ldots, p \tag{3.10}$$

where δ is some small positive number, say 10^{-6}. The objective function $Q_n(\hat{\theta}_k)$ in equation (3.2) should be evaluated at each iteration to ensure that a reduction in its value has been obtained.

3.3.2 Newton-Raphson Method

The basic idea of Newton-Raphson method is to approximate the objective function in each iteration by a quadratic function and then move the current point to the turning point of the quadratic curve. This method has the advantage of potentially speeding up the convergence significantly, but has the possible disadvantage of making the algorithm more unstable.

The objective function $Q_n(\theta)$ in (3.2) can be approximated as

$$Q_n(\theta) \approx Q_n(\hat{\theta}_0) + G_0'(\theta - \hat{\theta}_0) + \frac{1}{2}(\theta - \hat{\theta}_0)' H_0 (\theta - \hat{\theta}_0) \qquad (3.11)$$

where $\hat{\theta}_0$ is a starting value and

$$G_0 = \frac{\partial Q_n(\hat{\theta}_0)}{\partial \theta} = -2\sum_{i=1}^{n}(y_i - f(x_i, \hat{\theta}_0))\frac{\partial f(x_i, \hat{\theta}_0)}{\partial \theta}$$

$$H_0 = \frac{\partial^2 Q_n(\hat{\theta}_0)}{\partial \theta \partial \theta'} = 2\sum_{i=1}^{n}\frac{\partial f(x_i, \hat{\theta}_0)}{\partial \theta}\frac{\partial f(x_i, \hat{\theta}_0)}{\partial \theta'} - 2\sum_{i=1}^{n}(y_i - f(x_i, \hat{\theta}_0))\frac{\partial^2 f(x_i, \hat{\theta}_0)}{\partial \theta \partial \theta'}$$

where H_0 is the Hessian Matrix and G_0 is the gradient vector of $Q_n(\theta)$.

The next value $\hat{\theta}_1$ of the Newton-Raphson iteration is obtained by minimizing the right-hand side of the approximation(3.11), which yields

$$\hat{\theta}_1 = \hat{\theta}_0 - H_0^{-1} G_0 \qquad (3.12)$$

This provides a natural basis for an iterative procedure for minimizing $Q_n(\theta)$. In general, we have at the kth iteration

$$\hat{\theta}_{k+1} = \hat{\theta}_k - H_k^{-1} G_k. \tag{3.13}$$

The iteration above is to be repeated until the sequence $\hat{\theta}_k$ is converged.

3.3.3 Steepest Descent Method

The method of steepest descent is one of the oldest optimization techniques. It is based on the simple principle that from a starting value $\hat{\theta}_0$ the best direction to go is the one that produces the largest local change in the steepest descent. The direction is defined by the gradient vector at the given $\hat{\theta}$. Hence the algorithm becomes

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k \frac{\partial Q_n(\hat{\theta}_k)}{\partial \theta}, \quad k = 0, 1, \cdots, \tag{3.14}$$

where k is the iteration count, $\hat{\theta}_k$ is the value of the kth iterate , and α_k is the step size which regulates how large a step the algorithm takes. Obviously, too large or too small steps may prevent the algorithm to converge, even if the steps are in the correct directions.

3.3.4 Grid Search Method

The direct grid search method relies only on evaluating $Q_n(\theta)$ at a sequence of points $\hat{\theta}_1, \hat{\theta}_2, \cdots$ and comparing values of $Q_n(\hat{\theta}_1)$, $Q_n(\hat{\theta}_2)$, $\cdots$, in order to reach the optimal values of θ (Walsh, 1975). The direct search method

can commonly handle the cases that the derivatives of $\partial Q_n(\theta)/\partial\theta$ are not continuous, or the Hessian Matrix is exact or near singular.

Grid search method is to construct a mess of grid points, and evaluate the objective function $Q_n(\theta)$ at each of these points. If the permissible range of θ is limited by

$$\mathbf{l} \leq \theta \leq \mathbf{u},$$

where $\mathbf{l}$ is a $p \times 1$ vector of the lower bounds and $\mathbf{u}$ is a $p \times 1$ vector of the upper bounds. All the grid points would be generated from a uniform distribution with lower bound $\mathbf{l}$ and upper bound $\mathbf{u}$.

The value of the objective function $Q_n(\theta)$ is evaluated at each point, and compared with the least value found before. If it is less than the least value, it replaces it and is retained; if it is greater, it is rejected. By taking the smallest value of $Q_n(\theta)$ in the permissible range of θ, we can obtain the optimal values of θ.

3.3.5 Comparisons of the Methods

There are two major disadvantages associated with Gauss-Newton and Newton-Raphson method. First, they all require **starting values**. In most of the simulation studies for estimation of nonlinear mixed effects models, as the true parameter values are known, they are used as starting values to avoid potential numerical optimization problems. However, Ge, Bickel and Rice (2003) mentioned that by starting at the true parameter values, a poor al-

gorithm may be favored, if it tends to stop early. If some values that are far away from true values are chosen as starting values instead, the iterative estimation algorithms converge very slowly or may even fail to converge at all. Moreover, by using the poor starting values, a local optimum may be obtained over a global one. Therefore, without enough prior knowledge about the possible true parameter values, it is extremely difficult to choose appropriate starting values. Second, they require the **computation of gradients or Hessian matrix**. If $Q_n(\theta)$ is not differentiable or not continuous, the gradients of $Q_n(\theta)$ are impossible to be calculated or expensive to derive. For the cases of high dimensions, the evaluation of gradients may be complicated, and the Hessian Matrix may be exact or near singular.

For the steepest descent method, it is simple and usually works best when the starting point is a long way from the optimum. Despite widespread use and formal convergence of the steepest descent algorithm, it is relative inefficient and seldom converges reliably.

For the random grid search method, there are several main advantages. If the problem dimension is not too large, the algorithm can perform fairly quickly. This approach does not require staring values, because the random grid search process will find the optimum value from the grid points. It is a derivative free method, and thus it greatly reduces the computing cost. Furthermore, this algorithm has a great generality and basically can be applied to any objective function. The grid search method is also associated with

several weaknesses. It can only be used in a small region near the optimal points, otherwise a large number of unwanted functions are calculated. If the original mesh is large, it is possible to miss the global minimum and choose a local minimum. In addition, information about $Q_n(\theta)$ obtained in the previous calculations is not being used to speed up the search for the optimal point. There are some auxiliary algorithms which can facilitate and improve the performance of the simple random grid search methods (Fu and Wang , 2002).

Chapter 4

Finite Sample Properties of the Second-order Least Squares Estimators

In this chapter, we conduct some simulation studies to demonstrate how the numerical computation can be done for the second-order least squares estimators (SLSE) using the statistical programming language R, and to investigate the finite sample properties of the SLSE.

4.1 Design of Simulation Studies

Quadratic, exponential and logistic models given in Wang (2005), are used in the simulation studies with different combinations of fixed and random effects. Here is a list of the models in the simulation studies:

Model 1. Quadratic model with two independent random effects

Model 2. Quadratic model with two dependent random effects

Model 3. Exponential model with one random effect

Model 4. Exponential model with two independent random effects

Model 5. Exponential model with two dependent random effects

Model 6. Logistic model with one random effect

Model 7. Logistic model with two independent random effects

All the random effects and random errors are generated from normal distributions.

In the design of simulation studies, N, the number of units, and n, the number of observations per unit, are taken as various values. For each of the sample sizes $R = 500$ Monte Carlo replicates are carried out and the Monte Carlo mean estimates and the simulation standard errors (SSE) for the estimators are computed. All the computations are conducted in R 2.1.0 on an IBM Workstation with a 2.2MHz CPU and 4GB RAM with standard hardware configuration.

As is frequently the case in fitting nonlinear mixed effects models by using statistical softwares, convergence, numerical complaints and numerical discrepancies problems will be encountered, because most of the global optimization procedures implemented in statistical softwares are based on iterative methods. The disadvantages of iterative methods have been discussed

in section 3.3.5. In practical programming, we have to flag out warning messages and treat false convergences as breakdowns to exclude those estimates. Because the grid search optimization method does not require starting values or the computation of gradients, it is implemented in the programming of our simulation studies to avoid those potential optimization problems involved in the iterative methods. In our simulation studies, the bounds of grid points for each parameter are set closed to the true parameter values, and a large number of grid points $m = 5000$ per parameter are used by considering computing time, and computational cost.

4.2 Quadratic Models

A general quadratic model can be written as

$$\begin{aligned} y_{ij} &= \xi_{1i} + \xi_{2i} x_{ij}^2 + \epsilon_{ij} \\ \xi_{1i} &= \theta_1 + \delta_{1i}, \quad \xi_{2i} = \theta_2 + \delta_{2i} \\ \delta_i &\sim N(0, \Sigma_\delta), \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2) \\ i &= 1, \ldots, N, \quad j = 1, \ldots, n \end{aligned} \tag{4.1}$$

where θ_1 and θ_2 are fixed effects; δ_{1i} and δ_{2i} are random effects; the variance-covariance matrix of the random effects is given as

$$\Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 & \sigma_{\delta 12} \\ \sigma_{\delta 21} & \sigma_{\delta 2}^2 \end{bmatrix}$$

The first two conditional **moments** are given in Wang (2005)

$$\mu_{ij}(\psi) = \theta_1 + \theta_2 x_{ij}^2 \tag{4.2}$$

$$\nu_{ijk}(\psi) = (\theta_1 + \theta_2 x_{ij}^2)(\theta_1 + \theta_2 x_{ij}^2) + \sigma_{ijk} + \sigma_{\delta 1}^2 + \sigma_{\delta 2}^2 x_{ij}^2 x_{ik}^2 + \sigma_{12}(x_{ij}^2 + x_{ik}^2) \tag{4.3}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

A set of simulated data for $N = 7$ units and $n = 5$ observations per unit is plotted in Figure 4.1, where the profile of each unit has a similar shape but with some noticeable within-unit and intra-unit variation. There is also a nonlinear dependence of the response on some parameters of interest.

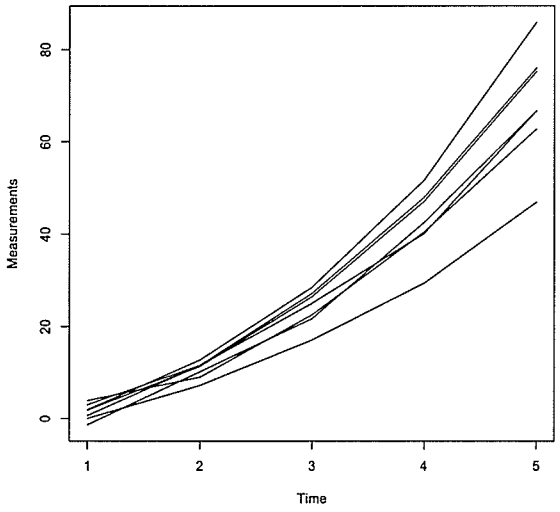

Figure 4.1: A set of simulated quadratic model data with $n = 5$, $N = 7$

The model is simulated based on the normality assumptions of random

effects. It is simulated in two different scenarios with **(1)** two independent random effects; **(2)** two dependent random effects.

4.2.1 Model 1: Two Independent Random Effects

A quadratic model with two independent random effects is considered. The data are generated using the following **parameter configuration**:

$$\begin{aligned} \theta_1 &= -20, \quad \theta_2 = 10, \quad \sigma_\epsilon^2 = 1, \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 6 & \sigma_{\delta 12} = 0 \\ \sigma_{\delta 21} = 0 & \sigma_{\delta 2}^2 = 3 \end{bmatrix} \\ x_j &= 1, \ldots, n \end{aligned}$$

Based on equations (4.2) and (4.3), the first two conditional **moments** of y_{ij} given x_i are respectively

$$\begin{aligned} \mu_{ij}(\psi) &= \theta_1 + \theta_2 x_{ij}^2 \\ \nu_{ijk}(\psi) &= (\theta_1 + \theta_2 x_{ij}^2)(\theta_1 + \theta_2 x_{ik}^2) + \sigma_{\delta 1}^2 + \sigma_{\delta 2}^2 x_{ij}^2 x_{ik}^2 + \sigma_{ijk} \end{aligned}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

Simulation results are reported in Table 4.1, 4.2 and 4.3.

Table 4.1: Quadratic model with 2 independent random effects, $n = 5$

$n = 5$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = -20$			
SLSE	-20.0659	-20.0354	-19.9738	-20.0374	-19.9965	-20.0108
	(0.0410)	(0.0328)	(0.0244)	(0.0188)	(0.0151)	(0.0117)
			$\theta_2 = 10$			
SLSE	10.0297	10.0110	9.9995	9.9856	10.0001	9.9955
	(0.0300)	(0.0196)	(0.0146)	(0.0109)	(0.0087)	(0.0057)
			$\sigma^2_{\delta 1} = 6$			
SLSE	6.0352	6.0650	5.9555	6.1145	5.9971	5.8652
	(0.0517)	(0.0524)	(0.0522)	(0.0499)	(0.0517)	(0.0507)
			$\sigma^2_{\delta 2} = 3$			
SLSE	3.0615	2.9408	3.0794	3.0538	3.0372	3.0019
	(0.0521)	(0.0516)	(0.0495)	(0.0503)	(0.0512)	(0.0507)
			$\sigma^2_{\epsilon} = 1$			
SLSE	1.0018	0.9803	1.0325	1.0273	1.0005	0.9990
	(0.0254)	(0.0260)	(0.0263)	(0.0259)	(0.0259)	(0.0254)

Table 4.2: Quadratic model with 2 independent random effects, $n = 10$

$n = 10$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = -20$			
SLSE	-20.0598	-19.9910	-20.0079	-20.0181	-20.0270	-19.9656
	(0.0428)	(0.0313)	(0.0251)	(0.0210)	(0.0277)	(0.0264)
			$\theta_2 = 10$			
SLSE	9.9449	9.9895	9.9909	10.0029	10.0018	9.9994
	(0.0291)	(0.0193)	(0.0139)	(0.0114)	(0.0079)	(0.0061)
			$\sigma^2_{\delta 1} = 6$			
SLSE	5.9678	5.9403	6.0003	5.9877	5.9712	5.9985
	(0.0522)	(0.0518)	(0.0503)	(0.0502)	(0.0511)	(0.0507)
			$\sigma^2_{\delta 2} = 3$			
SLSE	2.9328	3.0322	3.0429	2.9338	3.0184	2.9843
	(0.0511)	(0.0521)	(0.0531)	(0.0512)	(0.0512)	(0.0515)
			$\sigma^2_{\epsilon} = 1$			
SLSE	1.0428	1.0306	1.0325	1.0015	1.0321	0.9970
	(0.0251)	(0.0255)	(0.0254)	(0.0255)	(0.0251)	(0.0255)

Table 4.3: Quadratic model with 2 independent random effects, $n = 15$

$n = 15$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = -20$			
SLSE	-20.0603	-20.0185	-20.0272	-19.9989	-19.9933	-20.0051
	(0.0444)	(0.0398)	(0.0389)	(0.0359)	(0.0353)	(0.0340)
			$\theta_2 = 10$			
SLSE	9.9725	10.0009	9.9925	9.9750	10.0041	9.9898
	(0.0297)	(0.0202)	(0.0141)	(0.0112)	(0.0080)	(0.0065)
			$\sigma^2_{\delta 1} = 6$			
SLSE	5.9358	5.9823	5.91853	6.0329	6.0815	6.0027
	(0.0511)	(0.0523)	(0.0505)	(0.0535)	(0.0520)	(0.0511)
			$\sigma^2_{\delta 2} = 3$			
SLSE	2.9979	2.9522	2.9791	3.0853	3.0226	2.9991
	(0.0521)	(0.0516)	(0.0512)	(0.0518)	(0.0531)	(0.0512)
			$\sigma^2_{\epsilon} = 1$			
SLSE	1.0521	0.9767	0.9393	0.9868	0.9899	1.0035
	(0.0264)	(0.0258)	(0.0269)	(0.0258)	(0.0261)	(0.0256)

4.2.2 Model 2: Two Dependent Random Effects

A quadratic model with two dependent random effects is simulated. The data are generated using the following **parameter configuration**:

$$\begin{aligned} \theta_1 &= -20, \quad \theta_2 = 10, \quad \sigma_\epsilon^2 = 1 \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 6 & \sigma_{\delta 12} = 0.8 \\ \sigma_{\delta 21} = 0.8 & \sigma_{\delta 2}^2 = 3 \end{bmatrix} \\ x_j &= 1, \ldots, n \end{aligned}$$

Based on equations (4.2) and (4.3), the first two conditional **moments** of y_{ij} given x_i are respectively

$$\begin{aligned} \mu_{ij}(\psi) &= \theta_1 + \theta_2 x_{ij}^2 \\ \nu_{ijk}(\psi) &= (\theta_1 + \theta_2 x_{ij}^2)(\theta_1 + \theta_2 x_{ik}^2) + \sigma_{\delta 1}^2 + \sigma_{\delta 2}^2 x_{ij}^2 x_{ik}^2 + \\ &\quad \sigma_{12}^2 (x_{ij}^2 + x_{ik}^2) + \sigma_{ijk} \end{aligned}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

Table 4.4, 4.5 and 4.6 display the simulation results.

Table 4.4: Quadratic model with 2 dependent random effects, $n = 5$

$n = 5$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = -20$			
SLSE	-19.9952	-19.9965	-20.0051	-20.0067	-20.0365	-20.0124
	(0.0442)	(0.0314)	(0.0230)	(0.0213)	(0.0198)	(0.0176)
			$\theta_2 = 10$			
SLSE	10.0208	10.0125	9.9792	9.9815	9.9869	10.0034
	(0.0305)	(0.0203)	(0.0138)	(0.0110)	(0.0082)	(0.0060)
			$\sigma^2_{\delta 1} = 6$			
SLSE	5.9866	6.0685	6.1445	5.9367	6.0171	6.0661
	(0.0524)	(0.0508)	(0.0525)	(0.0526)	(0.0508)	(0.0510)
			$\sigma^2_{\delta 2} = 3$			
SLSE	3.0677	2.9291	3.0660	2.9671	3.1011	2.9772
	(0.0509)	(0.0516)	(0.0513)	(0.0512)	(0.0516)	(0.0506)
			$\sigma_{12} = 0.8$			
SLSE	0.8076	0.7901	0.8011	0.8050	0.8035	0.7989
	(0.0051)	(0.0053)	(0.0051)	(0.0052)	(0.0051)	(0.0051)
			$\sigma^2_{\epsilon} = 1$			
SLSE	1.0171	1.0178	0.9931	0.9647	1.0181	0.9748
	(0.0265)	(0.0258)	(0.0262)	(0.0254)	(0.0251)	(0.251)

Table 4.5: Quadratic model with 2 dependent random effects, $n = 10$

$n = 10$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = -20$			
SLSE	-19.9889	-20.0147	-20.0672	-20.0234	19.9976	20.0031
	(0.0435)	(0.0308)	(0.0233)	(0.0199)	(0.0190)	(0.0173)
			$\theta_2 = 10$			
SLSE	10.0295	9.9720	9.9841	9.9831	9.9958	10.0001
	(0.0293)	(0.0203)	(0.0145)	(0.0114)	(0.0078)	(0.0056)
			$\sigma^2_{\delta 1} = 6$			
SLSE	5.9403	5.9665	6.0190	5.9931	5.9703	6.0151
	(0.0518)	(0.0511)	(0.0512)	(0.0508)	(0.0513)	(0.0512)
			$\sigma^2_{\delta 2} = 3$			
SLSE	2.9100	3.0195	2.9772	3.0690	2.9681	3.0343
	(0.0512)	(0.0510)	(0.0519)	(0.0521)	(0.0511)	(0.0512)
			$\sigma_{12} = 0.8$			
SLSE	0.7962	0.7960	0.8007	0.8000	0.7969	0.8005
	(0.0052)	(0.0052)	(0.0052)	(0.0052)	(0.0051)	(0.0052)
			$\sigma^2_{\epsilon} = 1$			
SLSE	1.0013	1.0368	0.9646	0.9778	0.9959	1.0081
	(0.0257)	(0.0265)	(0.0255)	(0.0251)	(0.0253)	(0.0255)

Table 4.6: Quadratic model with 2 dependent random effects, $n = 15$

$n = 15$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = -20$			
SLSE	-20.0580	-20.0033	-19.9958	20.0014	19.9921	20.0071
	(0.0435)	(0.0343)	(0.0271)	(0.0208)	(0.0197)	(0.0181)
			$\theta_2 = 10$			
SLSE	10.0263	9.9741	9.9973	9.9854	10.0100	10.0102
	(0.0283)	(0.0191)	(0.0140)	(0.0116)	(0.0073)	(0.0056)
			$\sigma^2_{\delta 1} = 6$			
SLSE	5.9820	5.9673	5.9651	5.9595	5.9904	5.9926
	(0.0512)	(0.0507)	(0.0504)	(0.0512)	(0.0506)	(0.0506)
			$\sigma^2_{\delta 2} = 3$			
SLSE	3.0691	2.9876	2.9963	3.0532	2.9789	2.9813
	(0.0515)	(0.0512)	(0.0508)	(0.0511)	(0.0515)	(0.0511)
			$\sigma_{12} = 0.8$			
SLSE	0.8076	0.8043	0.8036	0.7991	0.8034	0.7939
	(0.0051)	(0.0052)	(0.0052)	(0.0052)	(0.0052)	(0.0052)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9915	1.0668	0.9870	1.0051	0.9893	0.9921
	(0.0257)	(0.0251)	(0.0257)	(0.253)	(0.255)	(0.0250)

4.2.3 Summary of Simulation Results

Based on the simulation studies for these two quadratic models, the results can be summarized as:

(1). For finite sample sizes, the SLSE performs reasonably well for the quadratic models with two random effects.

(2). No apparent finite sample biases are noticed from the Monte Carlo mean estimates.

(3). For fixed effects, there is a clear pattern of decreasing SSE as the increase of sample sizes.

(4). The SSEs of random effects are fairly stable with the increase of sample sizes.

(5). The relative variabilities of the estimates of fixed effects are smaller than those of the random effects.

4.3 Exponential Models

An exponential model is considered here for the simulation. The model can be written as

$$\begin{aligned} y_{ij} &= \xi_{1i}\exp(\xi_{2i}x_{ij}) + \epsilon_{ij} \qquad (4.4)\\ \xi_{1i} &= \theta_1 + \delta_{1i}, \quad \xi_{2i} = \theta_2 + \delta_{2i} \\ \delta_i &\sim N(0, \Sigma_\delta), \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2) \\ i &= 1, \ldots, N, \quad j = 1, \ldots, n \end{aligned}$$

where θ_1 and θ_2 are fixed effects; δ_{1i} and δ_{2i} are random effects; the variance-covariance matrix of the random effects is given as

$$\Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 & \sigma_{\delta 12} \\ \sigma_{\delta 21} & \sigma_{\delta 2}^2 \end{bmatrix}$$

The first two conditional **moments** are given in Wang (2005)

$$\begin{aligned} \mu_{ij}(\psi) &= (\theta_1 + \sigma_{\delta 12}x_{ij})\exp(\theta_2 x_{ij} + x_{ij}^2\sigma_{\delta 2}^2/2) \qquad (4.5)\\ \nu_{ijk}(\psi) &= [\sigma_{\delta 1}^2 + (\theta_1 + \sigma_{\delta 12}(x_{ij} + x_{ik})^2] \times \\ &\quad \exp[(x_{ij} + x_{ik})\theta_2 + (x_{ij} + x_{ik})^2\sigma_{\delta 2}^2/2] + \sigma_{ijk} \qquad (4.6) \end{aligned}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

A set of simulated data for $N = 7$ units and $n = 5$ observations per unit is plotted in Figure 4.2. As we can see that all units have a similar shape curve but with noticeable within-group and intra-group variation. The response depends on parameters of interest nonlinearly.

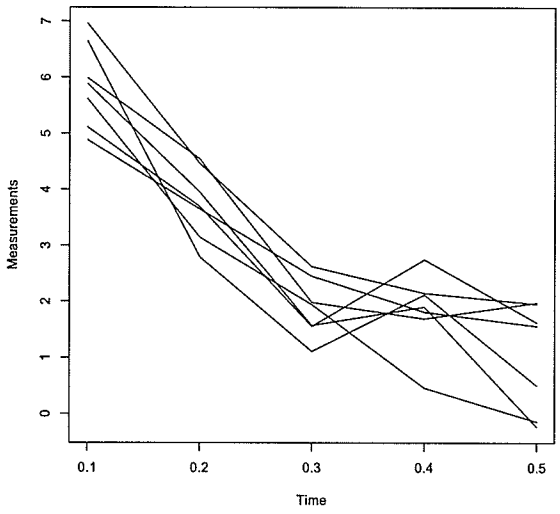

Figure 4.2: A set of simulated exponential model data with $n = 5$, $N = 7$

The model is simulated based on the normality assumption of random effects. It is generated in three different cases with **(1)** one random effect; **(2)** two independent random effects; **(3)** two dependent random effects.

4.3.1 Model 3: One Random Effect

An exponential model with one random effect is used to generate the data. Based on equation (4.4), the model is given by

$$\begin{aligned} y_{ij} &= \gamma \exp(\xi_i x_{ij}) + \epsilon_{ij} \\ \xi_i &= \theta_1 + \delta_{1i} \end{aligned}$$

where γ is the regression coefficient.

Based on equations (4.5) and (4.6), the first two conditional **moments** of y_{ij} in an exponential model with one random effect can be obtained as

$$\begin{aligned} \mu_{ij}(\psi) &= \gamma \exp(\theta_1 x_{ij} + x_{ij}^2 \sigma_\delta^2/2) \\ \nu_{ijk}(\psi) &= \gamma^2 exp(\theta_1(x_{ij} + x_{ik}) + (x_{ij} + x_{ik})^2 \sigma_\delta^2/2) + \sigma_{ijk} \end{aligned}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

The data have been generated using the following **parameter configuration**:

$$\begin{aligned} \gamma &= 10, \quad \theta = -0.5, \quad \sigma_\epsilon^2 = 1, \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 0.5 & \sigma_{\delta 12} = 0 \\ \sigma_{\delta 21} = 0 & \sigma_{\delta 2}^2 = 0 \end{bmatrix} \\ x_j &= 0.1, \ldots, 0.1n \end{aligned}$$

The simulation results for the fixed and random effects are summarized in Table 4.7, 4.8 and 4.9.

Table 4.7: Exponential model with one random effect, $n = 5$

$n = 5$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\gamma = 10$			
SLSE	10.0743	10.0334	10.0299	10.0176	10.0128	9.9909
	(0.0440)	(0.0298)	(0.0224)	(0.0179)	(0.0131)	(0.0098)
			$\theta_1 = -5$			
SLSE	-5.0534	-5.0196	-5.0142	-5.0009	-5.0086	-4.9910
	(0.0263)	(0.0195)	(0.0136)	(0.0113)	(0.0087)	(0.0063)
			$\sigma^2_{\delta 1} = 0.5$			
SLSE	0.4875	0.4987	0.4977	0.5085	0.5078	0.5092
	(0.0053)	(0.0053)	(0.0052)	(0.0052)	(0.0052)	(0.0053)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9863	0.9946	0.9857	1.0035	0.9968	0.9965
	(0.0053)	(0.0049)	(0.0051)	(0.0051)	(0.0050)	(0.0048)

Table 4.8: Exponential model with one random effect, $n = 10$

$n = 10$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\gamma = 10$			
SLSE	10.1139	10.0813	10.0419	10.0275	9.9869	10.0034
	(0.0397)	(0.0305)	(0.0216)	(0.0164)	(0.0115)	(0.0092)
			$\theta_1 = -5$			
SLSE	-5.0718	-5.0587	-5.0170	-5.0113	-4.9964	-5.004
	(0.0256)	(0.0188)	(0.0129)	(0.0098)	(0.0073)	(0.0057)
			$\sigma^2_{\delta 1} = 0.5$			
SLSE	0.4913	0.4988	0.4999	0.4996	0.4935	0.5006
	(0.0054)	(0.0054)	(0.0052)	(0.0052)	(0.0052)	(0.0051)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9846	0.9957	0.9998	1.0044	0.9861	1.0020
	(0.0050)	(0.0049)	(0.0047)	(0.0046)	(0.0048)	(0.0048)

Table 4.9: Exponential model with one random effect, $n = 15$

$n = 15$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\gamma = 10$			
SLSE	10.0375	9.9938	10.022	10.0072	10.0119	9.9989
	(0.0427)	(0.0297)	(0.0202)	(0.0166)	(0.0118)	(0.0083)
			$\theta_1 = -5$			
SLSE	-5.0346	-5.0247	-5.0274	-5.0138	-5.0053	-4.9985
	(0.0263)	(0.0183)	(0.0129)	(0.0100)	(0.0076)	(0.0051)
			$\sigma^2_{\delta 1} = 0.5$			
SLSE	0.4980	0.4941	0.4989	0.5010	0.5004	0.5006
	(0.0055)	(0.0054)	(0.0053)	(0.0052)	(0.0051)	(0.0052)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9893	0.9996	0.9949	1.0021	0.9962	1.0008
	(0.0047)	(0.0047)	(0.0045)	(0.0044)	(0.0045)	(0.0044)

4.3.2 Model 4: Two Independent Random Effects

An exponential model with two independent random effects is considered. Based on equation (4.4), the model can be written as

$$\begin{aligned} y_{ij} &= \xi_{1i} \exp(\xi_{2i} x_{ij}) + \epsilon_{ij} \\ \xi_{1i} &= \theta_1 + \delta_{1i}, \quad \xi_{2i} = \theta_2 + \delta_{2i} \end{aligned}$$

The first two conditional **moments** of y_{ij} in an exponential model with two independent random effects can be obtained based on equations (4.5) and (4.6)

$$\begin{aligned} \mu_{ij}(\psi) &= \theta_1 \exp(\theta_2 x_{ij} + x_{ij}^2 \sigma_{\delta 2}^2/2) \\ \nu_{ijk}(\psi) &= (\sigma_{\delta 1}^2 + \theta_1^2) \exp(\theta_2 (x_{ij} + x_{ik}) + (x_{ij} + x_{ik})^2 \sigma_{\delta 2}^2/2) + \sigma_{ijk} \end{aligned}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

The data are simulated using the following **parameter configuration**:

$$\begin{aligned} \theta_1 &= 10, \quad \theta_2 = -5, \quad \sigma_\epsilon^2 = 1, \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 1 & \sigma_{\delta 12} = 0 \\ \sigma_{\delta 21} = 0 & \sigma_{\delta 2}^2 = 0.5 \end{bmatrix} \\ x_j &= 1, \ldots, n \end{aligned}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

The simulation results for the fixed and random effects are summarized in Table 4.10, 4.11 and 4.12.

Table 4.10: Exponential model with two independent random effects, $n = 5$

$n = 5$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 10$			
SLSE	10.0075	10.0139	10.0415	9.9657	10.0311	10.0180
	(0.0431)	(0.0340)	(0.0233)	(0.0192)	(0.0103)	(0.0103)
			$\theta_2 = -5$			
SLSE	-5.0251	-5.0156	-5.0146	-4.9956	-5.0236	-5.0121
	(0.0262)	(0.0203)	(0.0140)	(0.0112)	(0.0086)	(0.0060)
			$\sigma^2_{\delta 1} = 1$			
SLSE	0.9961	1.0065	0.9916	0.9972	0.9958	1.0023
	(0.0052)	(0.0052)	(0.0051)	(0.0053)	(0.0051)	(0.0051)
			$\sigma^2_{\delta 2} = 0.5$			
SLSE	0.4938	0.4982	0.4982	0.4938	0.5058	0.5041
	(0.0051)	(0.0053)	(0.0051)	(0.0052)	(0.0051)	(0.0050)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9947	0.9963	0.9971	1.0013	1.0031	1.0006
	(0.0050)	(0.0049)	(0.0049)	(0.0051)	(0.0049)	(0.0049)

Table 4.11: Exponential model with two independent random effects, $n = 10$

$n = 10$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 10$			
SLSE	10.0440	10.0189	10.0469	9.9975	9.9989	10.0157
	(0.0451)	(0.0303)	(0.0234)	(0.0189)	(0.0135)	(0.0100)
			$\theta_2 = -5$			
SLSE	-5.0440	-5.0238	-5.0331	-4.9999	-5.0011	-5.0173
	(0.0265)	(0.0179)	(0.0137)	(0.0108)	(0.0078)	(0.0062)
			$\sigma^2_{\delta 1} = 1$			
SLSE	0.9965	1.0068	0.9974	1.0032	0.9974	1.0037
	(0.0050)	(0.0051)	(0.0052)	(0.0052)	(0.0050)	(0.005)
			$\sigma^2_{\delta 2} = 0.5$			
SLSE	0.4926	0.4912	0.4996	0.4988	0.5008	0.5044
	(0.0055)	(0.0053)	(0.0050)	(0.0052)	(0.0053)	(0.0053)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9979	0.9924	0.9938	1.0009	0.9968	1.0010
	(0.0050)	(0.0050)	(0.0048)	(0.0050)	(0.0046)	(0.0047)

Table 4.12: Exponential model with two independent random effects, $n = 15$

$n = 15$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 10$			
SLSE	10.0704	10.1019	10.0108	9.9966	10.0305	9.9980
	(0.0451)	(0.0319)	(0.0221)	(0.0182)	(0.0133)	(0.0086)
			$\theta_2 = -5$			
SLSE	-5.0700	-5.0444	-5.0151	-5.0048	-5.0117	-5.0038
	(0.0251)	(0.0181)	(0.0125)	(0.0104)	(0.0077)	(0.0058)
			$\sigma^2_{\delta 1} = 1$			
SLSE	0.9974	0.9928	1.0081	0.9972	0.9894	1.0012
	(0.0050)	(0.0053)	(0.0053)	(0.0051)	(0.0051)	(0.0051)
			$\sigma^2_{\delta 2} = 0.5$			
SLSE	0.4905	0.5038	0.4990	0.5047	0.5036	0.4983
	(0.0055)	(0.0055)	(0.0054)	(0.0053)	(0.0052)	(0.0052)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9949	0.9974	1.0061	0.9948	1.0017	1.0008
	(0.0050)	(0.0048)	(0.0045)	(0.0048)	(0.0046)	(0.0045)

4.3.3 Model 5: Two Dependent Random Effects

An exponential model with two dependent random effects is used for simulation. The model is given by

$$\begin{aligned} y_{ij} &= \xi_{1i} \exp(\xi_{2i} x_{ij}) + \epsilon_{ij} \\ \xi_{1i} &= \theta_1 + \delta_{1i}, \quad \xi_{2i} = \theta_2 + \delta_{2i} \end{aligned}$$

Based on equations (4.5) and (4.6), the first two conditional **moments** of y_{ij} in an exponential model with two dependent random effects can be obtained as

$$\begin{aligned} \mu_{ij}(\psi) &= (\theta_1 + \sigma_{12}) \exp(\theta_2 x_{ij} + x_{ij}^2 \sigma_{\delta 2}^2/2) \\ \nu_{ijk}(\psi) &= (\sigma_{\delta 1}^2 + (\theta_1 + \sigma_{12}(x_{ij} + x_{ik}))^2) \exp(\theta_2 (x_{ij} + x_{ik}) + \\ &\quad (x_{ij} + x_{ik})^2 \sigma_{\delta 2}^2/2) + \sigma_{ijk} \end{aligned}$$

The data are simulated using the following **parameter configuration**:

$$\begin{aligned} \theta_1 &= 10, \quad \theta_2 = -5, \quad \sigma_\epsilon^2 = 1, \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 1 & \sigma_{\delta 12} = 0.4 \\ \sigma_{\delta 21} = 0.4 & \sigma_{\delta 2}^2 = 0.5 \end{bmatrix} \\ x_j &= 1, \ldots, n \end{aligned}$$

The simulation results for the fixed and random effects are summarized in Table 4.13, 4.14 and 4.15.

Table 4.13: Exponential model with two dependent random effects, $n = 5$

$n = 5$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 10$			
SLSE	10.1127	10.0486	10.0474	9.9697	10.0273	10.0030
	(0.0460)	(0.0329)	(0.0244)	(0.0191)	(0.0142)	(0.0104)
			$\theta_2 = -5$			
SLSE	-5.1088	-5.0406	-5.0238	-4.9930	-5.0151	-4.9999
	(0.0280)	(0.0197)	(0.0139)	(0.0112)	(0.0084)	(0.0063)
			$\sigma^2_{\delta 1} = 1$			
SLSE	1.0030	0.9920	1.0084	0.9990	1.0057	0.9962
	(0.0052)	(0.0052)	(0.0052)	(0.0053)	(0.0050)	(0.0051)
			$\sigma^2_{\delta 2} = 0.5$			
SLSE	0.4935	0.5011	0.5002	0.4937	0.5014	0.4906
	(0.0054)	(0.0053)	(0.0052)	(0.0051)	(0.0050)	(0.0052)
			$\sigma^2_{\delta 12} = 0.4$			
SLSE	0.3966	0.4009	0.4088	0.4049	0.4017	0.3937
	(0.0051)	(0.0050)	(0.0051)	(0.0051)	(0.0050)	(0.0052)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9935	0.9976	0.9973	0.9996	0.9922	1.0011
	(0.0051)	(0.0051)	(0.0050)	(0.0048)	(0.0049)	(0.0050)

Table 4.14: Exponential model with two dependent random effects, $n = 10$

$n = 10$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 10$			
SLSE	10.0658	10.0656	9.9940	10.0369	10.0009	10.0117
	(0.0427)	(0.0298)	(0.0231)	(0.0186)	(0.0131)	(0.0100)
			$\theta_2 = -5$			
SLSE	-5.0559	-5.0719	-5.016	-5.0253	-5.0002	-5.0023
	(0.0251)	(0.0180)	(0.0131)	(0.0104)	(0.0081)	(0.0060)
			$\sigma^2_{\delta 1} = 1$			
SLSE	1.0073	0.9974	1.0032	0.9909	1.0071	0.9942
	(0.0049)	(0.0053)	(0.0051)	(0.0052)	(0.0053)	(0.0052)
			$\sigma^2_{\delta 2} = 0.5$			
SLSE	0.4837	0.4979	0.4978	0.4977	0.4981	0.4977
	(0.0054)	(0.0053)	(0.0052)	(0.0051)	(0.0050)	(0.0053)
			$\sigma^2_{\delta 12} = 0.4$			
SLSE	0.4057	0.3991	0.4014	0.3997	0.3919	0.4007
	(0.0051)	(0.0052)	(0.0050)	(0.0052)	(0.0050)	(0.0050)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9935	0.9976	1.0073	1.0040	1.0007	1.0002
	(0.0051)	(0.0051)	(0.0048)	(0.0048)	(0.0048)	(0.0047)

Table 4.15: Exponential model with two dependent random effects, $n = 15$

$n = 15$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 10$			
SLSE	10.0851	10.0287	9.9951	10.0255	10.0111	9.9989
	(0.0435)	(0.0312)	(0.0232)	(0.0184)	(0.0128)	(0.0089)
			$\theta_2 = -5$			
SLSE	-5.0742	-5.0236	-5.0346	-5.0037	-5.0026	-5.0009
	(0.0246)	(0.0184)	(0.0135)	(0.0104)	(0.0078)	(0.0058)
			$\sigma^2_{\delta 1} = 1$			
SLSE	1.0009	0.9992	1.0097	0.9985	1.0047	0.9988
	(0.0053)	(0.0052)	(0.0052)	(0.0050)	(0.0052)	(0.0050)
			$\sigma^2_{\delta 2} = 0.5$			
SLSE	0.4897	0.5056	0.4992	0.5037	0.5038	0.4997
	(0.0054)	(0.0054)	(0.0053)	(0.0055)	(0.0051)	(0.0053)
			$\sigma^2_{\delta 12} = 0.4$			
SLSE	0.4102	0.3991	0.4014	0.3974	0.4030	0.4002
	(0.0051)	(0.0052)	(0.0050)	(0.0052)	(0.0053)	(0.0050)
			$\sigma^2_{\epsilon} = 1$			
SLSE	0.9977	1.0037	1.0064	0.9942	0.9983	1.0015
	(0.0051)	(0.0047)	(0.0046)	(0.0045)	(0.0044)	(0.0044)

4.3.4 Summary of Simulation Results

Based on the simulation studies for these three exponential models, we have similar findings:

(1). For finite sample sizes, the SLSE performs reasonably well for the quadratic models with two random effects.

(2). No apparent finite sample biases are noticed from the Monte Carlo mean estimates.

(3). For fixed effects, there is a clear pattern of decreasing SSE as the increase of sample sizes.

(4). The SSEs of random effects are fairly stable with the increase of sample sizes.

(5). The relative variabilities of the estimates of fixed effects are smaller than those of the random effects.

4.4 Logistic Models

Logistic models are studied in this section. A general logistic model can be written as

$$\begin{aligned} y_{ij} &= \frac{\xi_{1i}}{1+\exp(-(x_{ij}-\xi_{2i})/\gamma)} + \epsilon_{ij} \\ \xi_{1i} &= \theta_1 + \delta_{1i}, \quad \xi_{2i} = \theta_2 + \delta_{2i} \\ \delta_i &\sim N(0, \Sigma_\delta), \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2) \\ i &= 1, \ldots, N, \quad j = 1, \ldots, n \end{aligned} \tag{4.7}$$

where γ is the regression coefficient; θ_1 and θ_2 are fixed effects; δ_{1i} and δ_{2i} are random effects; the variance-covariance matrix of the random effects is given as

$$\Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 & \sigma_{\delta 12} \\ \sigma_{\delta 21} & \sigma_{\delta 2}^2 \end{bmatrix}$$

A set of simulated data for $N = 15$ units and $n = 10$ observations per unit is plotted in Figure 4.3. As we can see that all units have a similar shape curve but with noticeable within-group and intra-group variation. We can also observed that the response depend on parameters of interest nonlinearly.

The model is simulated based on the normality assumption of random effects. It is simulated in two different scenarios with **(1)** one random effect; **(2)** two independent random effects.

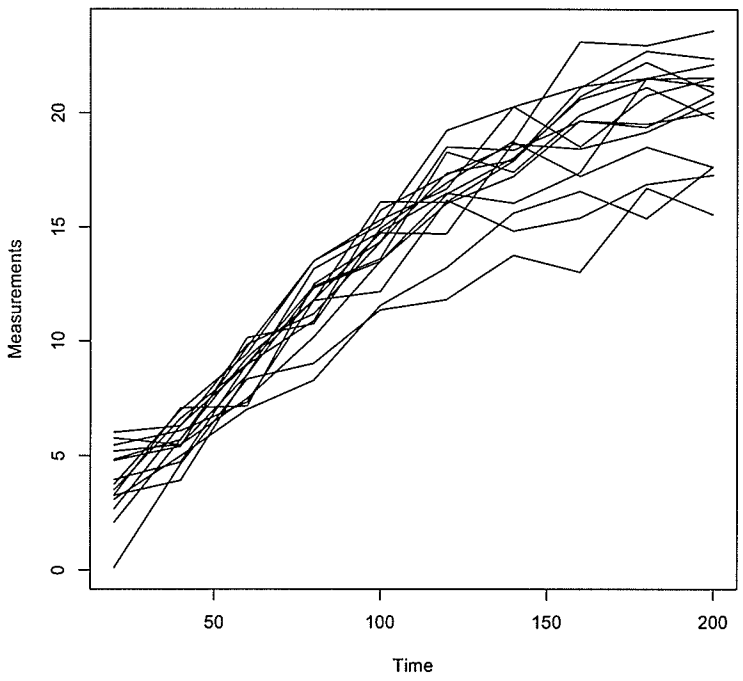

Figure 4.3: A set of simulated logistic model data with $n = 10$, $N = 15$

4.4.1 Model 6: One Random Effect

A logistic model with one random effect given in Wang (2005) is used to generate data. The model is given by

$$\begin{aligned} y_{ij} &= \frac{\xi_i}{1+\exp(-(x_{ij}-\gamma_1)/\gamma_2)} + \epsilon_{ij} \\ \xi_i &= \theta_1 + \delta_i \end{aligned}$$

The calculation of the first two conditional **moments** of y_{ij} given x_{ij} is straightforward and given in Wang (2005) as

$$\mu_{ij}(\psi) = \frac{\theta_1}{1+\exp(-(x_{ij}-\gamma_1)/\gamma_2)} \tag{4.8}$$

$$\nu_{ijk}(\psi) = \frac{\theta_1^2+\sigma_1^2}{(1+\exp(-(x_{ij}-\gamma_1)/\gamma_2))(1+\exp(-(x_{ik}-\gamma_1)/\gamma_2))} + \sigma_{ijk} \tag{4.9}$$

where $\sigma_{ijk} = \sigma_\epsilon^2$, if $j = k$, and zero otherwise.

The data are simulated using the following **parameter configuration**:

$$\theta_1 = 20, \quad \gamma_1 = 70, \quad \gamma_2 = 34, \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 9 & \sigma_{\delta 12} = 0 \\ \sigma_{\delta 21} = 0 & \sigma_{\delta 2}^2 = 0 \end{bmatrix}$$

$$x_j = 20,\ 40,\ \ldots,\ 20n$$

The simulation results for the fixed and random effects are summarized in Table 4.16, 4.17 and 4.18.

Table 4.16: Logistic model with one random effect, $n = 5$

$n = 5$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 20$			
SLSE	19.8818	19.9647	20.0466	20.0041	20.0372	19.9853
	(0.0510)	(0.0399)	(0.0317)	(0.0270)	(0.0212)	(0.0181)
			$\gamma_1 = 70$			
SLSE	69.9058	69.9359	70.2055	69.9968	70.0738	69.9569
	(0.0720)	(0.0715)	(0.0698)	(0.0672)	(0.0570)	(0.0540)
			$\gamma_2 = 34$			
SLSE	34.0462	34.0434	34.0556	33.9966	34.0279	33.9866
	(0.0592)	(0.0511)	(0.0432)	(0.0404)	(0.0338)	(0.0289)
			$\sigma_1^2 = 9$			
SLSE	9.0167	8.9831	8.9477	8.9891	9.0272	9.0286
	(0.0542)	(0.0517)	(0.0537)	(0.0506)	(0.0508)	(0.0517)
			$\sigma_\epsilon^2 = 1$			
SLSE	1.0140	0.9602	1.0091	0.9772	0.9766	0.9786
	(0.0215)	(0.0198)	(0.0180)	(0.0171)	(0.0229)	(0.0216)

Table 4.17: Logistic model with one random effect, $n = 10$

$n = 10$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 20$			
SLSE	19.8781	19.9772	20.0318	19.9662	19.9898	20.0007
	(0.0493)	(0.0352)	(0.0258)	(0.0196)	(0.0145)	(0.0103)
			$\beta_1 = 70$			
SLSE	69.9632	70.0966	70.0203	69.9680	70.0151	69.9839
	(0.0614)	(0.0508)	(0.0398)	(0.0334)	(0.0306)	(0.0216)
			$\beta_2 = 34$			
SLSE	33.9186	34.0490	34.0303	33.9557	33.9968	34.0125
	(0.0599)	(0.0493)	(0.0395)	(0.0315)	(0.0278)	(0.0219)
			$\sigma_1^2 = 9$			
SLSE	8.9293	9.0004	8.9625	8.9999	9.0556	9.0223
	(0.0508)	(0.0519)	(0.0528)	(0.0531)	(0.0516)	(0.0525)
			$\sigma_\epsilon^2 = 1$			
SLSE	0.9529	0.9961	1.0016	1.0196	1.0166	1.0401
	(0.0236)	(0.0234)	(0.0249)	(0.0236)	(0.0252)	(0.0250)

Table 4.18: Logistic model with one random effect, $n = 15$

$n = 15$	$N = 7$	$N = 15$	$N = 30$	$N = 50$	$N = 100$	$N = 200$
			$\theta_1 = 20$			
SLSE	20.0157	20.0452	19.9844	19.9844	20.0116	19.9842
	(0.0475)	(0.0355)	(0.0249)	(0.0191)	(0.0138)	(0.0096)
			$\beta_1 = 70$			
SLSE	70.0038	70.0106	69.9938	70.0546	69.9948	70.0084
	(0.0538)	(0.0427)	(0.0324)	(0.0294)	(0.0232)	(0.0164)
			$\beta_2 = 34$			
SLSE	34.0299	33.9601	33.9862	33.9768	34.0503	34.0094
	(0.0556)	(0.0426)	(0.0320)	(0.0283)	(0.0238)	(0.0151)
			$\sigma_1^2 = 9$			
SLSE	8.9889	9.0158	9.0072	9.0178	9.0281	9.0762
	(0.0544)	(0.0506)	(0.0523)	(0.0504)	(0.0518)	(0.0525)
			$\sigma_\epsilon^2 = 1$			
SLSE	0.9835	1.0202	0.9934	0.9814	0.9687	0.9952
	(0.0247)	(0.0240)	(0.0244)	(0.0253)	(0.0253)	(0.0245)

4.4.2 Model 7: Two Independent Random Effects

A logistic model with two random effects is used to generate data. The same model is also used by Pinherio and Bates (1995) for comparing different approximation maximum likelihood approaches. The model is given by

$$\begin{aligned} y_{ij} &= \frac{\xi_{1i}}{1+\exp(-(x_{ij}-\xi_{2i})/\gamma)} + \epsilon_{ij} \\ \xi_i &= \theta_1 + \delta_{1i}, \quad \xi_{2i} = \theta_2 + \delta_{2i} \end{aligned} \tag{4.10}$$

where γ is the regression coefficient; θ_1 and θ_2 are fixed effects; δ_{1i} and δ_{2i} are random effects; the variance-covariance matrix of the random effects is given as

$$\Sigma_\delta = \begin{bmatrix} \sigma^2_{\delta 1} & \sigma_{\delta 12} \\ \sigma_{\delta 21} & \sigma^2_{\delta 2} \end{bmatrix}$$

For this model, the first two conditional moments are difficult to derive; therefore, we use the simulation-based estimation (SBE) method to approximate the first two conditional moments.

The first two conditional moments are derived as

$$\begin{aligned} \mu_{ij,1}(\psi) &= \frac{1}{S}\frac{S_1 S_2}{\sigma_{\delta 1}\sigma_{\delta 2}} \sum_{s=1}^{S} \frac{u_{i_1 s}}{1+\exp(-(x_{ij}-u_{i_2 s})/\gamma)} \times \\ &\quad \exp\left(-\frac{(u_{i_1 s}-\theta_1)^2}{2\sigma^2_{\delta 1}} + \frac{u^2_{i_1 s}}{2S_1^2} - \frac{(u_{i_2 s}-\theta_2)^2}{2\sigma^2_{\delta 2}} + \frac{u^2_{i_2 s}}{2S_2^2}\right) \end{aligned}$$

$$\mu_{ij,2}(\psi) = \frac{1}{S}\frac{S_1 S_2}{\sigma_{\delta 1}\sigma_{\delta 2}} \sum_{s=S+1}^{2S} \frac{u_{i_1 s}}{1+\exp(-(x_{ij}-u_{i_2 s})/\gamma)} \times \exp\left(-\frac{(u_{i_1 s}-\theta_1)^2}{2\sigma_{\delta 1}^2} + \frac{u_{i_1 s}^2}{2S_1^2} - \frac{(u_{i_2 s}-\theta_2)^2}{2\sigma_{\delta 2}^2} + \frac{u_{i_2 s}^2}{2S_2^2}\right)$$

$$\nu_{ijk,1}(\psi) = \frac{1}{S}\frac{S_1 S_2}{\sigma_{\delta 1}\sigma_{\delta 2}} \sum_{s=1}^{S} \frac{u_{i_1 s}^2}{[1+\exp(-(x_{ij}-u_{i_2 s})/\gamma)][1+\exp(-(x_{ik}-u_{i_2 s})/\gamma)]} \times \exp\left(-\frac{(u_{i_1 s}-\theta_1)^2}{2\sigma_{\delta 1}^2} + \frac{u_{i_1 s}^2}{2S_1^2} - \frac{(u_{i_2 s}-\theta_2)^2}{2\sigma_{\delta 2}^2} + \frac{u_{i_2 s}^2}{2S_2^2}\right)$$

$$\nu_{ijk,2}(\psi) = \frac{1}{S}\frac{S_1 S_2}{\sigma_{\delta 1}\sigma_{\delta 2}} \sum_{s=S+1}^{2S} \frac{u_{i_1 s}^2}{[1+\exp(-(x_{ij}-u_{i_2 s})/\gamma)][1+\exp(-(x_{ik}-u_{i_2 s})/\gamma)]} \times \exp\left(-\frac{(u_{i_1 s}-\theta_1)^2}{2\sigma_{\delta 1}^2} + \frac{u_{i_1 s}^2}{2S_1^2} - \frac{(u_{i_2 s}-\theta_2)^2}{2\sigma_{\delta 2}^2} + \frac{u_{i_2 s}^2}{2S_2^2}\right)$$

The data are simulated using the following **parameter configuration**:

$$\theta_1 = 200, \quad \theta_2 = 700, \quad \gamma = 350, \quad \Sigma_\delta = \begin{bmatrix} \sigma_{\delta 1}^2 = 100 & \sigma_{\delta 12} = 0 \\ \sigma_{\delta 21} = 0 & \sigma_{\delta 2}^2 = 625 \end{bmatrix}$$

$$x_j = 20,\ 40,\ \ldots,\ 20n$$

$$u_{i_1 s} \sim N(200, 9^2), \quad u_{i_2 s} \sim N(700, 9^2), \quad S = 500$$

Because of the implementation of high dimensional matrix in the programming, this program runs relatively slow. Therefore, only two sample size configurations $n = 5$ $N = 7$, and $n = 10$ $N = 30$ are considered. $R = 500$

runs are carried out. With the increased RAM of computer, the running time of the program will be significantly reduced.

Table 4.19: Logistic model with two independent random effects $n = 5$, $N = 7$ and runs $R = 500$

	$\theta_1 = 200$	$\theta_2 = 700$	$\beta = 350$	$\sigma^2_{\delta 1} = 100$	$\sigma^2_{\delta 2} = 625$	$\sigma^2_{\varepsilon} = 25$
SBE	199.3850	699.3057	349.8222	104.8866	634.3594	25.3303
	(0.5984)	(0.56201)	(0.5896)	(0.0088)	(0.0533)	(0.2605)

Table 4.20: Logistic model with two independent random effects $n = 10$, $N = 30$ and runs $R = 500$

	$\theta_1 = 200$	$\theta_2 = 700$	$\beta = 350$	$\sigma^2_{\delta 1} = 100$	$\sigma^2_{\delta 2} = 625$	$\sigma^2_{\varepsilon} = 25$
SBE	199.1341	701.0309	350.3208	104.8797	634.3097	25.3549
	(0.5775)	(0.5933)	(0.6065)	(0.0106)	(0.0517)	(0.2959)

The simulation-based estimators should be generally less efficient than the second-order least squares estimators. As it is proved by Wang (2005), the efficiency loss caused by simulation decreases at rate $O(1/S)$. Therefore, with the increase of S, it will become more efficient.

4.4.3 Summary of Simulation Results

Here is a summary of the simulation results.

(1). For the finite sample sizes, both SLSE and SBE perform reasonably well for the logistic models.

(2). No apparent finite sample biases are noticed from the Monte Carlo mean estimates.

(3). For fixed effects, there is a clear pattern of decreasing SSE as the increase of sample sizes in Model 6.

(4). The SSEs of random effects are fairly stable with the increase of sample sizes in Model 6.

4.5 Conclusions

Through these simulation studies, we have demonstrated how the numerical computation can be done for the second-order least squares estimation method. Based on the simulation results for these seven different models, we can conclude that for finite sample sizes, the second-order least squares estimation approach performs reasonably well. All Monte Carlo mean estimates are close to the true parameter values. No apparent biases are noticed from the estimates. The simulation standard errors of fixed effects are decreasing with the increase of sample sizes. For example in Table 4.17, when $n = 10$ and $N = 7, 15, 30, 50, 100, 200$, the estimates of fixed effects θ_1 are 19.8781, 19.9772, 20.0318, 19.9662, 19.9898, and 20.0007, and are all very closed to the true value $\theta_1 = 20$; with the increase of N, the simulation standard error is decreasing from (0.0493), (0.0352), (0.0258), (0.0196), (0.0145), to (0.0103). Thus, the estimates are becoming more precise.

The simulation standard errors of random effects are fairly stable with the increase of sample sizes. For example in Table 4.8, the simulation standard errors of the estimates for the random effect are (0.0055), (0.0054), (0.0053), (0.0052), (0.0051), and (0.0052). We also notice that the relative variabilities of the fixed effects estimates are smaller than those of the random effects. These are not surprising because the estimates of random effects are usually more difficult to estimate and known to have fairly large standard deviations, especially when the sample sizes are relatively small.

Chapter 5

Non-normal Random effects

As introduced in chapter 1 and 2, the likelihood approach relies on the **normality** assumption of random effects. However, this assumption may not be realistic. Because the random effects are not observed, it may be difficult to verify this normality assumption (Mallet 1986; Davidian and Gallant 1992, 1993; Fattinger et al. 1995; Hartford and Davidian 1999). It is thus natural to be concerned whether these methods yield reliable results when the normality assumption is not appropriate. The consequences of misspecifying the distributions of random effects have been discussed for linear mixed effects models (Butler and Louis 1992; Verbeke and Lesaffre 1997 ; Tao et. al. 1999) and for nonlinear mixed effects models (Hartford and Davidian 1999). However, Wang's (2005) second-order least squares approach does **not** rely on the normal assumptions of the random effects. It can produce reliable estimates under non-normal random effects. Moreover, it is computational feasible.

In order to verify how well the second-order least squares estimation approach performs with non-normal random effects, three simulation studies are conducted; thereby, we also demonstrate how the numerical computation can be done generally. The second-order least squares estimates are compared with the ones obtained from Lindstrom and Bates' (1990) `nlme()`.

5.1 Design of Simulation Studies

In the simulation studies, an exponential model with one random effect is used. The random effect is generated from three non-normal distributions. Due to the effects of small sample sizes on estimation results, three relative large sample sizes are investigated at here, where the number of units, $n = 10$ and the number of observations per unit, $N = 15, 30, 50$. $R = 500$ Monte Carlo replicates are carried out. The Monte Carlo mean estimates and the corresponding simulation standard errors are reported. All computations are conducted in `R` 2.1.0 on an IBM Workstation with a 2.2MHz CPU and 4GB RAM with standard hardware configurations.

A direct random grid search method is applied for the minimization involved in the second-order least squares estimation algorithm. A reasonably large number of grid points $m = 5000$ per parameter are used by considering computing time and computational cost. The `nlme()` library of functions in `R` 2.1.0 implementing Lindstrom and Bates' (1990) linearization algorithm introduced in section 2.1 is used to generate likelihood estimation results.

Detailed description of the `nlme()` library can be found in Pinheiro and Bates (2000). For model fitting with `nlme()`, starting values are needed. Because Lindstrom and Bates' (1990) linearization algorithm has been well validated, the starting values are chosen as the true parameter values.

The random effect δ is generated from three different non-normal distributions.

(i). An **asymmetric** distribution for δ is attained by a mixture of two normal distributions, where $\delta \sim (1-\alpha)N(\mu, \sigma^2) + \alpha N(-\mu, \sigma^2)$, with mixing proportion $\alpha = 0.3$ and specified value of μ. The mean and variance of δ are respectively

$$\begin{aligned} \mu_\delta &= (1-2\alpha)\mu = 0.4\mu \\ \sigma_\delta^2 &= \sigma^2 + 4\alpha(1-\alpha)\mu^2 = \sigma^2 + 0.84\mu^2 \end{aligned}$$

A plot of the asymmetric distribution is shown in Figure 5.1

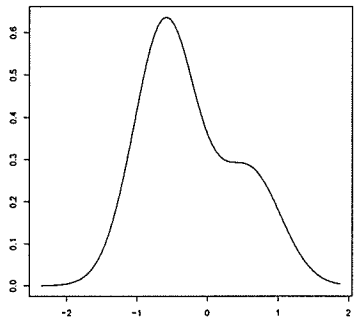

Figure 5.1: Asymmetric random effect

(ii). A **bimodal** distribution of δ is obtained by a mixture of two normal distributions, where $\delta \sim (1-\alpha)N(\mu, \sigma^2) + \alpha N(-\mu, \sigma^2)$, with mixing proportion $\alpha = 0.5$ and specified value of μ. The mean and variance of δ are respectively

$$\begin{aligned} \mu_\delta &= (1-2\alpha)\mu = 0 \\ \sigma_\delta^2 &= \sigma^2 + 4\alpha(1-\alpha)\mu^2 = \sigma^2 + \mu^2 \end{aligned}$$

A plot of the bimodal distribution is shown in Figure 5.2

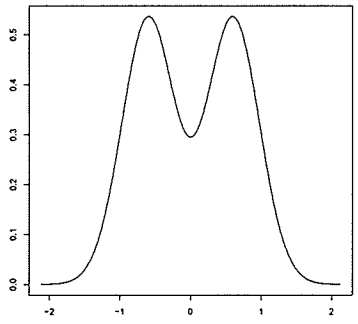

Figure 5.2: Bimodal random effect

(iii). A highly skewed **gamma** distribution with specified values of $\alpha = 1$ and $\beta = \sqrt{0.5}$ is used to simulate the random effect, where $\delta \sim G(\alpha, \beta)$. The mean and variance of δ are respectively

$$\mu_\delta = \alpha\beta, \quad \sigma_\delta^2 = \alpha\beta^2$$

A plot of the gamma distribution is shown in Figure 5.3

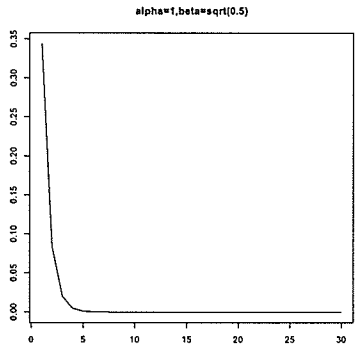

Figure 5.3: Gamma random effect

5.2 Exponential Model

Based on equation (4.4), an exponential model with one random effect is given by

$$\begin{aligned} y_{ij} &= \gamma \exp(\xi_i x_{ij}) + \epsilon_{ij} \\ \xi_i &= \theta_1 + \delta_i \end{aligned}$$

5.2.1 Asymmetric Random Effect

In the first scenario, the random effect is generated from an asymmetric distribution, where $\delta \sim (1-\alpha)N(\mu, \sigma^2) + \alpha N(-\mu, \sigma^2)$ and $\alpha = 0.3$. The first two conditional moments of y_{ij} are easy to derive

$$\begin{aligned}
\mu_{ij}(\psi) &= \gamma \exp(\theta_1 x_{ij})[(1-\alpha)\exp(\mu x_{ij} + x_{ij}^2\sigma_\delta^2/2) + \alpha \exp(-\mu x_{ij} + x_{ij}^2\sigma_\delta^2/2)] \\
\nu_{ijk}(\psi) &= \gamma^2 exp(\theta_1(x_{ij}+x_{ik}))[(1-\alpha)\exp(\mu(x_{ij}+x_{ik}) + (x_{ij}+x_{ik})^2\sigma_\delta^2/2) + \\
&\quad \alpha \exp(-\mu(x_{ij}+x_{ik}) + (x_{ij}+x_{ik})^2\sigma_\delta^2/2)] + \sigma_{ijk} \qquad (5.1)
\end{aligned}$$

The data have been generated using the following **parameter configuration**:

$$\begin{aligned}
\gamma &= 10, \quad \theta = -0.5, \quad \sigma_\delta^2 = 0.5, \quad \sigma_\epsilon^2 = 1, \quad \mu = 0.6 \\
x_j &= 0.1, \ldots, 0.1n
\end{aligned}$$

The Monte Carlo mean estimation results and the simulation standard errors are reported in Table 5.1 and 5.2

Table 5.1: Exponential model with one asymmetric random effect $n = 10$, $N = 30$

	$\gamma = 10$	$\theta = -5$	$\sigma_\delta^2 = 0.5$	$\sigma_\epsilon^2 = 1$
SLSE	10.0102	-4.7618	0.4976	0.9956
	(0.4748)	(0.3797)	(0.1161)	(0.1072)
`nlme()`	9.7846	-4.7982	0.4720	0.9976
	(0.5098)	(0.3412)	(0.2156)	(0.0847)

Table 5.2: Exponential model with one asymmetric random effect $n = 10$, $N = 50$

	$\gamma = 10$	$\theta = -5$	$\sigma_\delta^2 = 0.5$	$\sigma_\epsilon^2 = 1$
SLSE	10.0261	-4.7754	0.4963	0.9969
	(0.3796)	(0.3221)	(0.1154)	(0.1062)
`nlme()`	9.8324	-4.8234	0.5075	0.9996
	(0.3717)	(0.2712)	(0.1624)	(0.0655)

5.2.2 Bimodal Random Effect

In the second scenario, the random effect is generated from a bimodal distribution, where $\delta \sim (1-\alpha)N(\mu, \sigma^2) + \alpha N(-\mu, \sigma^2)$ and $\alpha = 0.5$. The first two conditional moments of y_{ij} are the same as equations in (5.1)

The data have been generated using the following **parameter configuration**:

$$\begin{aligned} \gamma &= 10, \quad \theta = -0.5, \quad \sigma_\delta^2 = 0.5, \quad \sigma_\epsilon^2 = 1, \quad \mu = 0 \\ x_j &= 0.1, \ldots, 0.1n \end{aligned}$$

The Monte Carlo mean estimation results and the simulation standard errors are reported in Table 5.3 and 5.4

Table 5.3: Exponential model with one bimodal random effect $n = 10$, $N = 30$

	$\gamma = 10$	$\theta = -5$	$\sigma_\delta^2 = 0.5$	$\sigma_\epsilon^2 = 1$
SLSE	10.0399	-5.0210	0.5103	0.9968
	(0.4793)	(0.2831)	(0.1188)	(0.1110)
`nlme()`	9.8395	-4.8185	0.4521	0.9963
	(0.4954)	(0.3211)	(0.2147)	(0.0832)

Table 5.4: Exponential model with one bimodal random effect $n = 10$, $N = 50$

	$\gamma = 10$	$\theta = -5$	$\sigma_\delta^2 = 0.5$	$\sigma_\epsilon^2 = 1$
SLSE	10.0548	-5.0219	0.5028	0.9983
	(0.3804)	(0.2299)	(0.1195)	(0.1073)
`nlme()`	9.8226	-4.8108	0.4477	1.0022
	(0.3958)	(0.2805)	(0.1696)	(0.0708)

5.2.3 Gamma Random Effect

In the third scenario, the random effect is generated from a gamma distribution, where $\delta \sim G(\alpha, \beta)$. The first two conditional moments of y_{ij} are derived based on the moment generating function of a gamma distribution.

$$\begin{aligned}
\mu_{ij}(\psi) &= E_i(\gamma \exp(\theta + \delta)x_{ij}) = \gamma \exp(\theta x_{ij})/(1 - \beta x_{ij})^\alpha \\
\nu_{ijk}(\psi) &= E(\gamma \exp(\theta + \delta)x_{ij}\gamma \exp(\theta + \delta)x_{ik}) \\
&= \gamma^2 exp(\theta(x_{ij} + x_{ik}))/(1 - \beta(x_{ij} + x_{ik})^\alpha)
\end{aligned}$$

The data have been generated using the following **parameter configuration**:

$$\begin{aligned}
\gamma &= 10, \quad \theta = -0.5, \quad \sigma_\delta^2 = 0.5, \quad \sigma_\epsilon^2 = 1, \quad \alpha = 1, \quad \beta = 0.5^{1/2} \\
x_j &= 0.1, \ldots, 0.1n
\end{aligned}$$

The Monte Carlo mean estimation results and the simulation standard

errors are reported in Table 5.5 and 5.6

Table 5.5: Exponential model with one gamma random effect $n = 10$, $N = 30$

	$\gamma = 10$	$\theta = -5$	$\sigma_\delta^2 = 0.5$	$\sigma_\epsilon^2 = 1$
SLSE	10.0361	-5.0471	0.4984	1.0024
	(0.5299)	(0.4381)	(0.1258)	(0.1149)
`nlme()`	9.8478	-4.8395	0.6343	0.9984
	(0.5437)	(0.4311)	(0.4086)	(0.0839)

Table 5.6: Exponential model with one gamma random effect $n = 10$, $N = 50$

	$\gamma = 10$	$\theta = -5$	$\sigma_\delta^2 = 0.5$	$\sigma_\epsilon^2 = 1$
SLSE	9.9936	-5.0060	0.4940	1.0067
	(0.5143)	(0.4280)	(0.1243)	(0.1135)
`nlme()`	9.8463	-4.8289	0.6678	0.9894
	(0.5389)	(0.4721)	(0.3544)	(0.0068)

5.2.4 Summary of Simulation Results

From Table 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6, some general simulation results are summarized.

(i). The regression coefficient and fixed effect

(a) Both methods are doing almost equally well for the estimation of the regression coefficient and fixed effect.

(b) For both methods, no apparent finite sample biases are noticed.

(ii). The random effect

(a) The SLS estimation method consistently generates estimates without any significant bias. All the SSEs are relative smaller than `nlme()`.

(b) There are some noticeable biases in estimates from `nlme()` method, except in the asymmetric case where $n = 10$ and $N = 50$. With increased deviations from the normality assumption, the SSE is increasing for the same sample size. For example, for $n = 10$ and $N = 50$, the SSE is equal to 0.1624 in the asymmetric case which is the smallest and the SSE is equal to 0.4086 in the gamma case which is the largest.

(iii). Random error

(a) For both methods, no apparent finite sample biases are noticed.

(b) The SSEs of the `nlme()` estimates are slightly smaller than the corresponding ones of SLSE.

5.3 Conclusions

From the simulation studies, we show how the numerical computation can be done for the second-order least squares estimators under the non-normal random effects.

Numerical findings from our limited simulation studies suggest that second-order least squares estimation algorithm gives consistent accurate estimates for both random and fixed effects when the random effect is not normal. Thus, we demonstrate the second-order least squares estimation algorithm does not rely on the normal assumptions of the random effects.

The `nlme()` method seems to give accurate and consistent estimates for the regression coefficient and fixed effect. Intuitively, this is not surprising because estimation of fixed effects is relatively robust to underlying assumptions (Hartford and Davidian, 1999). For the estimation of the random effect, the `nlme()` method seems to give increased bias with the increasing deviations from normal distributions (from asymmetric to bimodal to gamma distribution).

Chapter 6

Applications

In this section, the second-order least squares estimation approach will be applied to some real problems which have been extensively studied in the recent literature.

6.1 Orange Tree

In chapter 1, the growth model of orange tree is introduced. The data consist of seven measurements of the trunk circumference on each of five orange trees, taken over a period of 1600 days, originally presented by Draper and Smith (1981, p.524), and then used by Lindstrom and Bates (1990) as an illustration for estimation of nonlinear mixed effects models.

According to Lindstrom and Bates (1990), a logistic model is appropriate to fit the data. To account for the tree to tree variation, Lindstrom and Bates (1990) concluded in their analysis that only the asymptotic circumfer-

ence needs a random effect. Therefore, the following nonlinear mixed effects model was suggested

$$y_{ij} = \frac{\theta_1 + \delta_i}{1 + \exp(-(x_{ij} - \gamma_1)/\gamma_2)} + \epsilon_{ij}$$

where $i = 1, \cdots, 5, \quad j = 1, \cdots, 7, \quad \delta_i \sim N(0, \sigma_1^2), \quad \epsilon_{ij} \sim N(0, \sigma_2^2)$, and $x_j = 118, 484, 664, 1004, 1231, 1372, 1582$

The first two moments of this model are the same as equations in (4.8). Because true parameter values for the model are unknown, the estimates from second-order least squares method are compared with those from the maximum likelihood and the restricted maximum likelihood methods presented in Pinheiro and Bates (1995). The `nlme()` library of functions is used here to estimate the orange data.

From Table 6.1, we can see that the second-order least squares estimates are close to the ones obtained by the other two methods. The estimates for the random effect variance σ_1^2 look quite different, which is not surprising because its estimator is known to have a fairly large standard deviation.

Table 6.1: Estimation of the orange tree growth model

Parameters	θ_1	γ_1	γ_2	σ_1^2	σ_ϵ^2
SLSE	729.92	350.13	192.50	1002.41	61.00
MLE	727.91	348.07	192.05	1001.25	61.50
LME	722.56	344.16	191.05	990.29	61.56

6.2 A Pharmacodynamic Model

This nonlinear population pharmacodynamic model

$$y_{ij} = \theta_{1i} - \frac{\theta_{2i} x_{ij}}{\theta_{3i} + x_{ij}} + \epsilon_{ij} \tag{6.1}$$

was used by Walker (1996), for comparing the MLEs obtained with EM algorithm to approximate MLEs, and also used by Kuhn and Lavielle (2004) to compare the stochastic version of approximate EM (SAEM) algorithm with EM algorithm, first-order conditional estimation and Laplacian algorithms. According to Sheiner et al. (1991) and Walker (1996), the common application for this model is the analysis of blood pressure (y_{ij}) as a function of the dose (x_{ij}) of an antihypertensive drug from a longitudinal study. θ_{1i} represents a baseline response in the absence of treatment, θ_{2i} is the maximum effect of the drug, and θ_{3i} represents the dose which gives 50 percent of the maximum effect. The calculation of the first two moments for this pharmacodynamic model is fairly straightforward because all the random effects are independent. The first two conditional moments are given as

$$\mu_{ij}(\psi) = \theta_1 - \frac{\theta_2 x_{ij}}{\theta_3 + x_{ij}}$$

$$\nu_{ijk}(\psi) = \sigma_1^2 - \frac{\theta_1\theta_2\theta_3(x_{ij}+x_{ik}) + 2\theta_1\theta_2 x_{ij}x_{ik}}{\theta_3^2 + \theta_3(x_{ij}+x_{ik}) + x_{ij}x_{ik}} - \frac{\sigma_2^2 x_{ij}x_{ik}}{\sigma_3^2 + \theta_3(x_{ij}+x_{ik}) + x_{ij}x_{ik}} + \sigma_{ijk}$$

Because this example was used by way of a simulation study based on $i = 1, \cdots, 30$ and $j = 1, \cdots, 6$, the parameters were pre-set as

$$\theta_{1i} \sim_{i.i.d.} N(105, 64), \quad \theta_{2i} \sim_{i.i.d.} N(12, 36)$$

$$\theta_{3i} \sim_{i.i.d.} N(10, 12.25), \quad \epsilon_{ij} \sim_{i.i.d.} N(0, 4)$$

and $x_{i1} = 0, x_{i2} = 5, x_{i3} = 10, x_{i4} = 20, x_{i5} = 40, x_{i6} = 80$

The second-order least squares estimates for the parameters are calculated based on $R = 1000$ replications and 10000 grid values. The corresponding parameter estimations and simulation standard errors are given in table 6.2. We can see that all the estimates are very close to the real values and with relative small simulation standard errors.

Table 6.2: Estimation of the pharmadynamic model

	$\theta_1 = 105$	$\theta_2 = 12$	$\theta_3 = 10$	$\sigma_1^2 = 64$	$\sigma_2^2 = 36$	$\sigma_3^2 = 12.25$	$\sigma_\epsilon^2 = 4$
SLSE	104.1089	12.2108	10.3271	64.2630	35.8369	11.9032	4.0012
	(0.0553)	(0.0056)	(0.0189)	(0.0725)	(0.0731)	(0.0542)	(0.0360)

Chapter 7

Summary and Further Research

In statistical literature, the most popular estimation approach for nonlinear mixed effects models is the likelihood method. However, it is usually difficult to obtain a closed-form expression for the likelihood function, especially when the random effects are multi-dimensional. Moreover, most existing approximate likelihood approaches rely on the normality assumption of random effects. Wang (2005) proposed the second-order least squares estimation method which produces consistent estimators and does not rely on any parametric assumptions for the distributions of random effects. The potential computational issue of deriving the moment equations with multiple integrals has been addressed using the method of simulated moments.

We have performed several simulation studies for the second-order least squares estimation method proposed by Wang (2005), and applied it in two real data sets. From both the simulation studies and the real applications, we have demonstrated that how the second-order least squares estimators

can be numerical calculated, and how it can be implemented in a statistical computing language R. We have reported the results of our simulation studies undertaken to gain insight into the performance of the second-order least squares estimators. Although it is not appropriate to draw general conclusions from such limited simulation studies, the results suggest second-order least squares estimation method performs reasonably well for finite sample sizes when the random effects follow normal or even non-normal distributions.

Further research is required for the second-order least squares estimation method to find a more efficient estimator by involving a nonnegative definite weighting matrix in the objective function. It is natural to extend the second-order least squares estimation approach to the case where ϵ_{ij} and ϵ_{ik} are correlated. Moreover, the approach can be extended to the situation where the individuals have unbalanced observations.

Chapter 8

Appendix

8.1 Programs for Quadratic Models

8.1.1 Model 1: Two Independent Random Effects

```
"q2mde" <-
function(R, n, N, n0)
{
print(Sys.time())
cat("Simulating Quadratic Model with 2 Independent Random Effects","\n",
"Number of iterations:" ,R, "\n",
"Number of obs/group :" ,n, "\n",
"Number of groups:" ,N, "\n",
"Number of grid points:" ,n0, "\n \ n")
results <- matrix(0, R, 5)
x <- c(1:n)
```

```
x1 <- x*x
for (i in 1:(n-1)){x1 <- c(x1,x[i]*x[(i+1):n])}
# setting parameters#
fixed <- c(-20, 10)
rand <- 0.3*abs(fixed)
parameters <- c(fixed, rand, 1)
pa <- parameters
cov <- c(pa[3], 0, 0,pa[4])
#variance-covariance matrix#
Sigma <- matrix(cov,2,2)
for (j in 1:R)
{
#generate multivariate normal data#
ff <- mvrnorm(n=N, rep(0, 2), Sigma)
bb <- matrix(rep(ff, each=n1), n*N,2)
y <- (pa[1] + bb[,1]) + (pa[2]+bb[,2])*x^2 + rnorm(n*N, 0, sqrt(pa[5]))
yy <- matrix(y, N, n, byrow=TRUE)
a <- matrix(c(runif(n0,-22,-18), runif(n0,8,12), runif(n0,4,8),
runif(n0,1,5),runif(n0,0.5,1.5)), n0, 5)
m11 <- c(a[,1])%*%t(rep(1,n1)) + c(a[,2])%*%t(x^2)
m1 <- matrix(rep(m11,n2),n0, n*N)
Q1 <- apply((y-t(m1))*(y-t(m1)), 2, sum)
x2 <- m11*m11
```

```
for (i in 1:(n-1)) {  x2 <- cbind(x2,m11[,i]*m11[,(i+1):n1])}
m2 <- c(a[,3])%*%t(c(rep(1,(n1+1)*n1/2))) + x2 + c(a[,4])%*%t(x1^2)
m2[,1:n1] <- m2[,1:n1] + a[,5]
y2 <- yy*yy
for (i in 1:(n-1)) { y2 <- cbind(y2, yy[,i]yy[,(i+1):n]) }
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2), n0, N(n+1)n/2)
Q2 <- apply((y2-t(m2))*(y2-t(m2)), 2, sum)
Q3 <- Q1 + Q2
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
}
mde <- apply(results,2,mean)
ssd <- sqrt(apply(results,2,var)/N)
#format the output#
q.tabl<-data.frame(parameters,mde,ssd)
dimnames(q.tabl)<-list(c("Theata1","Theata2","Random.effect1",
"Random.effect2","Random.error"),
c("True Value","MDE","SSD"))
print(q.tabl)
Sys.time()
}
```

8.1.2 Model 2: Two Dependent Random Effects

```
"q2rmde" <-
function(R, n, N, n0)
{
print(Sys.time())
cat("Simulating Quadratic Model with 2 Dependent Random Effects","\n",
"Number of iterations:" ,R, "\n",
"Number of obs/group :" ,n, "\n",
"Number of groups:" ,N, "\n",
"Number of grid points:" ,n0, "\n \ n")
results <- matrix(0,N,6)
x <- c(1:n1)
x1 <- x*x
for (i in 1:(n1-1)){x1 <- c(x1,x[i]*x[(i+1):n1]) }
x3 <- x*x+x*x
for (i in 1:(n1-1)){x3 <- c(x3,x[i]*x[i]+x[(i+1):n1]*x[(i+1):n1])}
fixed <- c(-20, 10)
rand <- 0.3*abs(fixed)
parameters <- c(fixed, rand, 1, 0.8)
pa <- parameters
cov <- c(pa[3], pa[6], pa[6],pa[4])
Sigma <- matrix(cov,2,2) #variance-covariance matrix#
```

```
for (j in 1:N)
{
ff <- mvrnorm(n=n2, rep(0, 2), Sigma) #generate multivariate normal
data#
bb <- matrix(rep(ff, each=n1),n1*n2,2)
y <- (pa[1]+bb[,1]) + (pa[2]+bb[,2])*(x*x) + rnorm(n1*n2,0,sqrt(pa[5]))
yy <- matrix(y,n2,n1,byrow=TRUE)
a <- matrix(c(runif(n0,-22,-18), runif(n0,8,12), runif(n0,4,8),
runif(n0,1,5),runif(n0,0,2),runif(n0,0.6,1.0)), n0, 6)
m11 <- c(a[,1])%*%t(rep(1,n1)) + c(a[,2])%*%t(x*x)
m1 <- matrix(rep(m11,n2),n0,n1*n2)
Q1 <- apply((y-t(m1)(y-t(m1), sum)
x2 <- m11*m11
for (i in 1:(n1-1)){ x2 <-cbind(x2,m11[,i]*m11[,(i+1):n1]) }
m2 <- c(a[,3])%*%t(c(rep(1,(n1+1)*n1/2))) + x2 + c(a[,4])%*%t(x1*x1)
+c(a[,6])%*%t(x3)
m2[,1:n1] <- m2[,1:n1] + a[,5]
y2 <- yy*yy
for (i in 1:(n1-1)) { y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1]) }
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2),n0, n2*(n1+1)*n1/2)
Q2 <- apply((y2-t(m2))(y2-t(m2)),2,sum)
Q3 <- Q1 + Q2
```

```
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
} mde <- apply(results,2,mean)
ssd <-sqrt(apply(results,2,var)/N)
q.tabl<-data.frame(parameters,mde,ssd)
dimnames(q.tabl)<-list(c("Theata1","Theata2","Random.effect1",
"Random.effect2", "Random.error","Covirance"),
c("True Value","MDE","SSD"))
print(q.tabl)
Sys.time()
}
```

8.2 Programs for Exponential Models

8.2.1 Model 3: One Random Effect

```
"emde" <-
function(N,n1,n2,n0)
{
print(Sys.time())
cat("Simulating Exponential Model with one Random Effect","\ n",
"Number of iterations:" ,N, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n", "Number of grid points:" ,n0, " \
n \ n")
results <- matrix(0,N,4)
parameters <- c(10,-5,0.5,1)
pa <- parameters
x <- 0.1*c(1:n1)
x1 <- x+x
for (i in 1:(n1-1)) { x1 <- c(x1,x[i]+x[(i+1):n1]) }
for (j in 1:N)
{
rand <- rep(rnorm(n2,pa[2],sqrt(pa[3])),each=n1)
y <- pa[1]*exp(rand*x)+rnorm(n1*n2,0,sqrt(pa[4]))
yy <- matrix(y,n2,n1,byrow=TRUE)
```

```
a <- matrix(c(runif(n0,8,12), runif(n0,-7,-3), runif(n0,0.1,0.9),
runif(n0,0.1,1.5)), n0, 4)
m1 <- c(a[,1])%*%t(c(rep(1,n1))) *exp(c(a[,2])%*%t(x)+
c(a[,3])%*%t(x*x)/2)
m1 <- matrix(rep(m1,n2),n0,n1*n2)
Q1 <- apply((y-t(m1))(y-t(m1)), 2, sum)
m2 <- c(a[,1]*a[,1])%*%t(c(rep(1,(n1+1)*n1/2)))*exp(c(a[,2])%*%t(x1)
+c(a[,3])%*%t(x1*x1)/2)
m2[,1:n1] <- m2[,1:n1] + a[,4]
y2 <- yy*yy
for (i in 1:(n1-1))  y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1])
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2),n0, n2*(n1+1)*n1/2)
Q2 <- apply((y2-t(m2))(y2-t(m2)),2,sum)
Q3 <- Q1 + Q2
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
}
mde <- apply(results,2,mean)
ssd <- sqrt(apply(results,2,var)/N)
q.tabl<-data.frame(parameters,mde,ssd)
dimnames(q.tabl)<-list(c("Theata1","Fixed","Random.effect",
"Random.error"), c("True Value","MDE","SSD"))
```

```
print(q.tabl) Sys.time() }
```

8.2.2 Model 4: Two Independent Random Effects

```
"e2mde" <-
function(N,n1,n2,n0)
{
print(Sys.time()) cat("Simulating Exponential Model with 2 Independent
Random Effects","\ n",
"Number of iterations:" ,N, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n",
"Number of grid points:" ,n0, "\ n \ n")
results <- matrix(0,N,5)
x <- 0.1*c(1:n1)
x1 <- x+x
for (i in 1:(n1-1)) { x1 <- c(x1,x[i]+x[(i+1):n1]) }
parameters <- c(10,-5,1,0.5,1)
pa <- parameters
Sigma <- matrix(c(pa[3],0,0,pa[4]),2,2)
for (j in 1:N)
{
ff <- mvrnorm(n=n2, rep(0, 2), Sigma)
bb <- matrix(rep(ff, each=n1),n1*n2,2)
y <-(pa[1]+bb[,1])*exp((pa[2]+bb[,2])*x)+rnorm(n1*n2,0,sqrt(pa[5]))
```

```
yy <- matrix(y,n2,n1,byrow=TRUE)
a <- matrix(c(runif(n0,8,12), runif(n0,-7,-3), runif(n0,0.1,1.5),
runif(n0,0.1,1.5),runif(n0,0.1,1.5)), n0, 5)
m1 <- c(a[,1])%*%t(c(rep(1,n1)))*exp(c(a[,2])%*%t(x)+
c(a[,4])%*%t(x*x)/2)
m1 <- matrix(rep(m1,n2),n0,n1*n2)
Q1 <- apply((y-t(m1))(y-t(m1)), 2, sum)
m2 <- c(a[,1]*a[,1]+a[,3]a[,3])%*%t(c(rep(1,(n1+1)*n1/2)))*
exp(c(a[,2])%*%t(x1)+c(a[,4])%*%t(x1*x1)/2)
m2[,1:n1] <- m2[,1:n1] + a[,5]
y2 <- yy*yy for (i in 1:(n1-1))
{ y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1]) }
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2),n0, n2*(n1+1)*n1/2)
Q2 <- apply((y2-t(m2))(y2-t(m2)),2,sum)
Q3 <- Q1 + Q2
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
}
mde <- apply(results,2,mean)
ssd <- sqrt(apply(results,2,var)/N)
q.tabl <-data.frame(parameters,mde,ssd)
dimnames(q.tabl)<-list(c("Fixed1","Fixed2","Random.effect1",
```

```
"Random.effect2", "Random.error"), c("True Value","MDE","SSD"))
print(q.tabl)
Sys.time()
}
```

8.2.3 Model 5: Two Dependent Random Effects

```
"e2rmde" <-
function(N,n1,n2,n0)
{
print(Sys.time())
cat("Simulating Exponential Model with 2 Correlated
Random Effects","\ n",
"Number of iterations:" ,N, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n",
"Number of grid points:" ,n0, "\ n \ n")
results <- matrix(0,N,6)
x <- 0.1*c(1:n1)
x1 <- x+x for (i in 1:(n1-1)) { x1 <- c(x1,x[i]+x[(i+1):n1]) }
parameters <- c(10,-5,1,0.5,1,0.4)
pa <- parameters
Sigma <- matrix(c(pa[3],pa[6],pa[6],pa[4]),2,2)
for (j in 1:N)
{
ff <- mvrnorm(n=n2, rep(0, 2), Sigma)
bb <- matrix(rep(ff, each=n1),n1*n2,2)
y <-(pa[1]+bb[,1])*exp((pa[2]+bb[,2])*x)+rnorm(n1*n2,0,sqrt(pa[5]))
```

```
yy <- matrix(y,n2,n1,byrow=TRUE)
a <- matrix(c(runif(n0,8,12), runif(n0,-7,-3), runif(n0,0.8,1.2),
runif(n0,0.3,0.7),runif(n0,0.8,1.2),runif(n0,0.2,0.6)), n0, 6)
m1 <- (c(a[,1])%*%t(c(rep(1,n1)))+c(a[,6])%*%t(x)) *
exp(c(a[,2])%*%t(x)+ c(a[,4])%*%t(x*x)/2)
m1 <- matrix(rep(m1,n2),n0,n1*n2)
Q1 <- apply((y-t(m1))(y-t(m1)), 2, sum)
m2 <- (c(a[,3]*a[,3])%*%t(rep(1,(n1+1)*n1/2))+
(a[,1]%*%t(rep(1,(n1+1)*n1/2))+
c(a[,6])%*%t(x1*x1)))*exp(c(a[,2])%*%t(x1) +c(a[,4])%*%t(x1*x1)/2)
m2[,1:n1] <- m2[,1:n1] + a[,5]
y2 <- yy*yy
for (i in 1:(n1-1)) { y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1]) }
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2),n0, n2*(n1+1)*n1/2)
Q2 <- apply((y2-t(m2))(y2-t(m2)),2,sum)
Q3 <- Q1 + Q2
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
}
mde <- apply(results,2,mean)
ssd <- sqrt(apply(results,2,var)/N)
q.tabl <-data.frame(parameters,mde,ssd)
```

```
dimnames(q.tabl)<-list(c("Fixed1","Fixed2","Random.effect1",
"Random.effect2","Random.error", "Covirance"),
c("True Value","MDE","SSD"))
print(q.tabl)
Sys.time() }
```

8.3 Programs for Logistic Models

8.3.1 Model 6: One Random Effect

```
"lmde" <-
function(N, n1, n2, n0)
{
print(Sys.time())
cat("Simulating Logistic Model with 1 Random Effect","\ n",
"Number of iterations:" ,N, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n",
"Number of grid points:" ,n0, "\ n \n")
x <- c(seq(from=20,to=20*n1,by=20))
xx <- rep(seq(from=20,to=20*n1,by=20),n2)
pa <- c(20,70,34,9,1)
results <- matrix(0,N,5)
for (j in 1:N)
{
a <- matrix(c(runif(n0,18,22), runif(n0,pa[2]-2,pa[2]+2),
runif(n0,32,36), runif(n0,7,11), runif(n0,0.5,1.5)), n0, 5)
rr1 <- rep(rnorm(n2,pa[1],sqrt(pa[4])),each=n1)
y <- rr1/(1+exp((pa[2]-x)/pa[3]))+rnorm(n1*n2,0,sqrt(pa[5]))
yy <- matrix(y,n2,n1,byrow=TRUE)
```

```
deno <-(1+exp((c(a[,2]/a[,3])%*%t(c(rep(1,n1))))-c(1/a[,3])%*%t(x)))
m1 <- c(a[,1])%*%t(c(rep(1,n1)))/deno
m1 <- matrix(rep(m1,n2),n0,n1*n2)
Q1 <- apply((y-t(m1))(y-t(m1)),2,sum)
deno1 <- deno*deno for (i in 1:(n1-1))
{
deno1 <- cbind(deno1,deno[,i]*deno[,(i+1):n1])
}
m2 <- (c(a[,1]*a[,1])%*%t(c(rep(1,(n1+1)*n1/2)))+
c(a[,4])%*%t(c(rep(1,(n1+1)*n1/2))))/deno1
m2[,1:n1] <- m2[,1:n1] + a[,5]
y2 <- yy*yy
for (i in 1:(n1-1)) { y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1]) }
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2),n0, n2*(n1+1)*n1/2)
m2[,1:n1] <- m2[,1:n1] + a[,5]
Q2 <- apply((y2-t(m2))(y2-t(m2)),2,sum)
Q3 <- Q1 + Q2
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
}
mde <- apply(results,2,mean)
ssd <- sqrt(apply(results,2,var)/N)
```

```
q.tabl<-data.frame(pa,mde,ssd)
dimnames(q.tabl)<-list(c("Theata1","Theata2","Fixed",
"Random.effect","Random.error"), c("True Value","MDE","SSD"))
print(q.tabl)
Sys.time()
}
```

8.3.2 Model 7: Two Independent Random Effects

```
"logsmde2" <-
function (N,n1,n2, S,n0)
{
print(Sys.time())
cat("Simulating Logistic Model with 2 Random Effect","\ n",
"Number of iterations:" ,N, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n",
"Size of random sample:" ,n2*S, "\ n",
"Number of grid points:" ,n0, "\ n\ n")
result <- matrix(0,N,6)
x<-seq(from=200,to=200*n1,by=200)
pa <- c(200,700,350,100,625,25)
s0 <- 50 #for simulate moments#
SS <-2*S
one <- rep(1,n1)
for (I in 1:N)
{
Q <- matrix(0,n0,1)
b <- matrix(c(runif(n0,195,215), runif(n0,695,715), runif(n0,345,355),
runif(n0,95,105), runif(n0,620,630), runif(n0,20,30)), n0, 6)
```

```
rr1 <-rep(rnorm(n2,pa[1],sqrt(pa[4])),each=n1)
rr2 <- rep(rnorm(n2,pa[2],sqrt(pa[5])),each=n1)
y <- rr1/(1+exp((rr2-x)/pa[3]))+rnorm(n1*n2,0,sqrt(pa[6]))
yy <- matrix(y,n2,n1,byrow=TRUE)
u1 <- matrix(rnorm(n2*(SS),200,s0),n2,SS)
u2 <- matrix(rnorm(n2*(SS),200,s0),n2,SS)
z1 <- outer(u2,x,FUN="-")
tlcol <- (n1+1)*n1/2
ones <- rep(1,tlcol)
for (J in 1:n0)
{
cons <- sqrt(s0^4/(b[3*n0+J]*b[4*n0+J]))
v1 <-outer(u1*exp(-(u1-b[J])(u1-b[J])/(2*b[3*n0+J])+
u1*u1/(2*s0*s0)-(u2-b[n0+J])(u2-b[n0+J])/(2*b[4*n0+J])+
u2*u2/(2*s0*s0)),one)
vv1 <- 1+exp(z1/b[2*n0+J])
mm1 <- v1/vv1
M1 <- t(rbind(c(t(apply(mm1[,1:S,], c(1,3), sum))),
c(t(apply(mm1[,(S+1):SS,], c(1,3), sum)))))
m1 <- y-M1
Q1 <- t(m1[,1])%*%m1[,2]
N1=n1
v2 <-outer(u1*u1*exp(-(u1-b[J])(u1-b[J])/(2*b[3*n0+J])+u1*u1/(2*s0*s0)
```

```
-(u2-b[n0+J])(u2-b[n0+J])/(2*b[4*n0+J])+u2*u2/(2*s0*s0)),ones)
vv2 <- vv1*vv1
for (i in 1:(n1-1))
{ N1 <- N1+(n1-i)
vv2 <- array(c(vv2,array(vv1[,,i],c(SS,SS,n1-i))*
array(vv1[,,(i+1):n1], c(SS,SS,n1-i))),c(n2,SS,N1)) }
mm2 <- v2/vv2
M2 <- t(rbind(c(t(apply(mm2[,1:S,], c(1,3), sum))),
c(t(apply(mm2[,(S+1):SS,], c(1,3), sum)))))
y2 <-yy*yy-b[4*n0+J]
for (i in 1:(n1-1)) { y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1]) }
m2 <- c(y2)-M2
Q2 <- t(m2[,1])%*%m2[,2]
Q[J] <- cons*(Q1+Q2)
}
posit <- (Q>0)
Q <- Q[posit]
b <- b[posit,]
K <- which(Q==min(Q))
result[I,] <- b[K,]
}
mde <- apply(result,2,mean)
ssd <- sqrt(apply(result,2,var)/N)
```

```
log.tabl <- data.frame(pa,mde,ssd)
dimnames(log.tabl) <- list(c("Fixed.1","Fixed.2","Fixed.3",
"Random.effect.1","Random.effect.2","Random.error"),
c("True Value","MDE","SSD"))
print(log.tabl)
Sys.time()
}
```

8.4 Programs for Non-normal Random Effects Models

8.4.1 Maximum Likelihood Estimation

```
"emle.asy" <-
function(n,n1,n2)
{
print(Sys.time())
cat("Simulating Exponential Model with 1 Random Effects From bimodal
symmetric Distribution","\ n",
"Number of iterations:" ,n, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n\ n")
options(digits=6)
result1 <- matrix(0,n,4)
m1 <- 0 options(warn = 2)
parameters <- c(10,-5,0.5,1)
pa <- parameters p1 <- pa[1:2] #starting vlaue for nlme()#
x <- 0.1*rep(c(1:n1),n2)
mu0 <- -0.6
rv <- pa[3]
w0 <- 0.3
rmu <- (1-2*w0)*mu0
```

```
mixv <- rv -4*w0*(1-w0)*mu0*mu0
obj <- norMix(mu=c(mu0,-mu0), sig2 = c(mixv, mixv), w=c(1-w0,w0))
for (j in 1:n)
{
b2 <- rep(rnorMix(n2,obj),each=n1)-rmu+pa[2]
y <- pa[1]*exp(b2*x)+rnorm(n1*n2,0,sqrt(pa[4]))
z <- rep(1:n2,each=n1)
o < - data.frame(cbind(z,x,y))
oo <- groupedData(y x|z,data=o)
fn.exp <- try(nlme(y R1*exp(R2*x),
data=oo,
fixed=R1+R2 1,random=R2 1,
control=nlmeControl(maxIter=200, tolerance = 1e-06),
start=p1,method="ML"), TRUE)
if (inherits(fn.exp, "nlme")) # only for successful fits
result1[j,]<-matrix(c(fn.exp$coef$fixed,
as.numeric(VarCorr(fn.exp)[,1])),1,4)
}
result11 <- matrix(pa,1,4)
for ( j in 1:n)
{
if (result1[j,1]!=0) { m1 = m1+1
result11 <- rbind(result11,result1[j,])} }
```

```
result11 <-result11[2:(m1+1),]
print(m1)
mle <- apply(result11,2,mean)
ssd<-sqrt(apply(result11,2,var)/n)
truepa <- matrix(pa,m1,4,byrow=TRUE)
summse <- (result11-truepa)*(result11-truepa)
rmse <- sqrt(apply(summse,2,mean))
exp.tabl <- data.frame(pa,mle,ssd,rmse)
dimnames(exp.tabl)<-list(c("Fixed.1","Fixed.2","Random.effect",
"Random.error"), c("True Value","MLE","SSD","RMSE"))
print(exp.tabl)
Sys.time()
}
```

8.4.2 Second-order Least Squares Estimation

```
"eslse.bio" <-
function(N,n1,n2,n0)
{
print(Sys.time())
cat("Simulating Exponential Model with 1 Random Effects","\ n",
"Number of iterations:" ,N, "\ n",
"Number of obs/group :" ,n1, "\ n",
"Number of groups:" ,n2, "\ n",
"Number of grid points:" ,n0, "\ n \ n")
results <- matrix(0,N,4)
parameters <- c(10,-5,0.5,1)
pa <- parameters
x <- 0.1*c(1:n1)
x1 <- x+x for (i in 1:(n1-1)) { x1 <- c(x1,x[i]+x[(i+1):n1]) }
mu0 <- -0.6
rv <- pa[3]
w0 <- 0.5
rmu <- (1-2*w0)*mu0
mixv <- rv -4*w0*(1-w0)*mu0*mu0
obj <- norMix(mu=c(mu0,-mu0), sig2 = c(mixv, mixv), w=c(1-w0,w0))
for (j in 1:N)
```

```
{
rand <- rep(rnorMix(n2,obj),each=n1)-rmu+pa[2]
y <- pa[1]*exp(rand*x)+rnorm(n1*n2,0,sqrt(pa[4]))
yy <- matrix(y,n2,n1,byrow=TRUE)
a <- matrix(c(runif(n0,8,12), runif(n0,-7,-3),
runif(n0,0.3,0.7), runif(n0,0.8,1.2)), n0, 4)
rv1 <- a[,3] -4*w0*(1-w0)*mu0*mu0
m1 <- (1-w0)*c(a[,1])%*%t(c(rep(1,n1))) *exp(c(a[,2]+mu0)%*%t(x)+
c(rv1)%*%t(x*x)/2)+ w0*c(a[,1])%*%t(c(rep(1,n1))) *
exp(c(a[,2]-mu0)%*%t(x)+ c(rv1)%*%t(x*x)/2)
m1 <- matrix(rep(m1,n2),n0,n1*n2)
Q1 <- apply((y-t(m1))(y-t(m1)), 2, sum)
m2 <- (1-w0)*c(a[,1]*a[,1])%*%t(c(rep(1,(n1+1)*n1/2)))*
exp(c(a[,2]+mu0)%*%t(x1) +c(rv1)%*%t(x1*x1)/2)+w0*c(a[,1]*a[,1])%*
%t(c(rep(1,(n1+1)*n1/2)))*
exp(c(a[,2]-mu0)%*%t(x1)+c(rv1)%*%t(x1*x1)/2)
m2[,1:n1] <- m2[,1:n1] + a[,4]
y2 <- yy*yy for (i in 1:(n1-1))
{y2 <- cbind(y2,yy[,i]*yy[,(i+1):n1]) }
y2 <- c(t(y2))
m2 <- matrix(rep(m2,n2),n0, n2*(n1+1)*n1/2)
Q2 <- apply((y2-t(m2))(y2-t(m2)),2,sum)
Q3 <- Q1 + Q2
```

```
k <- which(Q3==min(Q3))
results[j,] <- a[k,]
}
mde <- apply(results,2,mean)
ssd <- sqrt(apply(results,2,var)/N)
truepa <- matrix(pa,N,4,byrow=TRUE)
summse <- (results-truepa)(results-truepa)
rmse <- sqrt(apply(summse,2,mean))
q.tabl<-data.frame(parameters,mde,ssd,rmse)
dimnames(q.tabl)<-list(c("Theata1","Fixed","Random.effect",
"Random.error"), c("True Value","MDE","SSD","RMSE"))
print(q.tabl)
Sys.time()
}
```

Bibliography

[1] Amemiya, T. (1985). *Advanced Econometrics.* Basil Blackwell, Cornwall.

[2] Bates, D. M. and Watts D. G. (1998). *Nonlinear Regression Analysis and Its Applications.* John Wiley and Sons, New York.

[3] Berkey, C. (1982). Bayesian approach for nonlinear growth model. *Biometrics,* 38, 953-961.

[4] Butler, S. M. and Louis, T. A. (1992). Random effects models with nonparametric priors. *Medicine of Statistics* , 11, 1981-2000.

[5] Concordet, D. and Nunez, O. G. (2002). A simulated pseudo-maximum likelihood estimator for nonlinear mixed models. *Journal of Computation Statistics and Data Analysis*, 39, 187-201.

[6] Das, S. and Krishen, A. (1999). Some bootstrap methods in nonlinear mixed-effect models. *Journal of Statistical Planning and Inference*, 75, 237-245.

[7] Davidian, M. and Giltinan, D. M. (1992). Smooth nonparametric maximum likelihood estimation for population pharmacokinetics, with application quindine. *Journal of Pharmacokinetics and Biopharmaceutics*, 20, 529 - 556.

[8] Davidian, M. and Giltinan, D. M. (1993). The nonlinear mixed effects model with a smooth random effects Density. *Biometrika*, 80, 475-488.

[9] Davidian, M. and Giltinan, D. M. (1995). *Nonlinear Models for Repeated Measurement data.* Chapman and Hall, London.

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

[11] Demidenko, E. (2004). *Mixed Models - Theory and Applications.* John Wiley and Sons, New Jersey.

[12] Dempster, A. P, Laird, N. M, and Rubin, D. B. (1997). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39 ,1977, 1-38.

[13] Dixon, L. W. (1972). *Nonlinear Optimization.* English Universities Press, London.

[14] Draper, N. R. and Simth, H. (1998). *Applied Regression Analysis*, 3rd edition., Wiley, New York.

[15] Fattinger, K. E., Sheiner, L. B. and Verotta, D. (1995). A new method to explore the distribution of interindividual random effects in nonlinear mixed effects models. *Biometrics*, 51, 1236-1251.

[16] Foulds, L. R. (1981). *Optimization Techniques: An Introduction.* Springer - Verlag, New York.

[17] Fu, J. C. and Wang, L. (2002). A random-discretization based Monte Carlo sampling method and its application. *Methodology and Computing in Applied Probability*, 4, 5-25.

[18] Gallant, R. (1987). *Nonlinear Statistical Models.* John Wiley and Sons, New York.

[19] Gallant, R., Ronald A. and Douglas W. N. (1987). Semi-Nonparametric Maximum Likelihood Estimation, *Econometrica*, 55, 363-390.

[20] Ge Z., Bickel, P. J. and Rice J. A. (2003). An approximate likelihood approach to nonlinear mixed effects models via spline approximation. *Journal of Computation Statistics and Data Analysis*, 46, 747-776.

[21] Goldstein, H. (1991). Nonlinear multilevel models with an application to discrete response data. *Econometrica*, 57, 1317-1339.

[22] Hartford, A. and Davidian, M. (2000). Consequences of misspecifying assumptions in nonlinear mixed effects models. *Computational Statistics and Data Analysis*, 34, 139-164.

[23] Jennrich, R. I. (1969). Asymptotic properties of nonlinear least squares estimation. *Annals of Mathematical Statistics*, 40, 633-643.

[24] Kuhn, E. and Lavielle, M. (2004). Maximum likelihood estimation in nonlinear mixed effects models. *Journal of Computation Statistics and Data Analysis*, 39, 187-201.

[25] Lai, L. T. and Shih C. M. (2003). Nonparametric estimation in nonlinear mixed effects models. *Biometrika*, 90, 1-13.

[26] Laid, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38, 963-974.

[27] Lee, S. Y. and Xu, L. (2004). Influence analyses of nonlinear mixed-effects models. *Journal of Computation Statistics and Data Analysis*, 45, 321-341.

[28] Liang, K. Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrica*, 73, 13-22.

[29] Lindsey, J. K. and Lambert, P. (1998). On the appropriateness of marginal models for repeated measurements in clinical trials. *Statistics in Medicine*, 17, 447-469.

[30] Lindstrom, M. J. and Bates, D. M. (1990). Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46, 673-687.

[31] Longford, N.T. (1994). Logistic regression with random coefficients. *Computational Statistics and Data Analysis*, 17, 1-15.

[32] Lu, J. C. and Meeker, W. Q. (1993). Using degradation measures to estimate a time-to-failure distribution. *Technometrics*. 35, 161-174.

[33] Mallet, A. (1986). A maximum likelihood estimation method for random coeffcient regression models. *Biometrika*, 73, 645-656.

[34] Mallet, A. (1992). A nonparametric method for population pharmacokinetics: A case study. *Biopharmaceutical Section, American Statistics Association*, 11-20.

[35] Mallet, A. and Mentre, F. (1994). Handling covariates in population pharmacokinetics. *International Journal of Biomedical Computation*, 36, 25-33.

[36] Mallet, A., Mentre, F., Steimer, J. and Lokiek, F. (1988). Nonparametric maximum likelihood estimation for population pharmacokinetics, with applications to cyclosporine. Jouranl of Pharmacokinetics and Biopharmeutics, 16, 311 - 327.

[37] Mentre, F. and Gomeni, R. (1995). A two step iterative algorithm for estimation in nonlinear mixed effect models with an evaluation in population pharmacokinetics. *Journal of Biopharameceutical Statistics*, 5, 141-158.

[38] Montgomery, D. C., Peck, E. A., and Vining G. G. (2001). *Introduction to Linear Regression Analysis.* John Wiley and Sons, New York.

[39] Pinheiro, J. C. and Bates, D. M. (1995). Approximation of the log-likelihood function in the nonlinear mixed elects models. *Journal of Computational Graphics Statistics*, 4, 12-35.

[40] Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-effects models in S and S-plus.* Springer, Berlin.

[41] Racine-Poon, A. (1985). A Bayesian approach to nonlinear random effects models. *Biometrics*, 41, 1015-1023.

[42] Ramos, R. Q. and Pantula, S. G. (1995). Estimation of nonlinear random coefficient models. *Journal of Statistics Probability Letter*, 24, 49-56.

[43] Renard, L., Sandres, P., and Delmas, J. M. (1996). Pharmacokinetic-pharmacodynamic model for spiramycin in staphylococcal mastitis. *Journal of veterinary pharmacology and therapeutics*, 19, 95-103.

[44] Sheiner, L. B. and Beal, S. L. (1980). Evaluation of methods for estimating population pharmacokinetic parameters. Michaelis - Menten Model : routine clinical pharmacokinetic data. *Journal of Pharmacokinectics and Biophamaceutics*, 8, 553-571.

[45] Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization.* John Wiley and Sons, New Jersey.

[46] Spector, P. (1994). *An Introduction to S and S-plus.* Duxbury Press, California.

[47] Steimer, J., Mallet, A., Golmard, J., and Boisieux, J. (1984). Alternative approaches to estimation of population pharmacokinetic parameters: Comparison with the nonlinear mixed-effect model. *Drug Metabolism Review,* 15, 265-292.

[48] Tao, H., Palta, M., Yandell, B. S. and Newton, M.A. (1999). An estimation method for the semiparametric mixed effects model. *Biometrics,* 55, 102-110.

[49] Tolle, H. (1975). *Optimization Methods.* Springer-Verlag, New York.

[50] Venables, W. N. and Ripley, B. D. (1997). *Modern applied statistics with S-plus.* Springer, Berlin.

[51] Verbeke, G. and Lesaffre, E. (1997). The effect of misspecifying the random-effects distribution in linear mixed models for longitudinal data. *Computational Statistics and Data Analysis,* 23, 541-556.

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

[53] Vonesh, E. F. and Carter, R. L. (1992). Mixed-effects nonlinear regression for unbalanced repeated measures. *Biometrics*, 48, 1-18.

[54] Vonesh, E. F. and Chinchilli, V. M. (1997). *Linear and Nonlinear Models for the Analysis of Repeated Measurements.* Marcel Dekker, New York.

[55] Walker, S. (1996). An EM algorithm for nonlinear random effects models. *Biometrics*, 52(3), 934-944.

[56] Walsh, G. R. (1975). *Methods of Optimization.* John Wiley and Sons, London.

[57] Wang, L. (2003). Estimation of Nonlinear Berkson-type Measurement Error Model. *Statistica Sinica*, 13, 1201-1210.

[58] Wang, L. (2004). Estimation of Nonlinear Models with Berkson Measurement Errors. *Annals of Statistics*, 32, 2559-2579.

[59] Wang, L. (2005). A unified approach to estimation of nonlinear mixed effects and Berkson measurement error models. Working Paper, Department of Statistics, University of Manitoba.

[60] Wolfinger, R. (1993). Laplace's approximation for nonlinear mixed models. *Biometrica*, 80, 791-795.