

Article

Stochastic Expectation Maximization Algorithm for Linear Mixed-Effects Model with Interactions in the Presence of Incomplete Data

Alandra Zakkour^{1,2}, Cyril Perret^{1,2} and Yousri Slaoui^{1,*} 

¹ Laboratoire de Mathématiques et Applications, Université de Poitiers, 11 Boulevard Marie et Pierre Curie, 86962 Futuroscope Chasseneuil, CEDEX 9, 86073 Poitiers, France

² CeRCA-CNRS UMR 7295, Université de Poitiers, 5 rue T. Lefebvre, MSHS, CEDEX 9, 86073 Poitiers, France

* Correspondence: yousri.slaoui@math.univ-poitiers.fr

Abstract: The purpose of this paper is to propose a new algorithm based on stochastic expectation maximization (SEM) to deal with the problem of unobserved values when multiple interactions in a linear mixed-effects model (LMEM) are present. We test the effectiveness of the proposed algorithm with the stochastic approximation expectation maximization (SAEM) and Monte Carlo Markov chain (MCMC) algorithms. This comparison is implemented to highlight the importance of including the maximum effects that can affect the model. The applications are made on both simulated psychological and real data. The findings demonstrate that our proposed SEM algorithm is highly preferable to the other competitor algorithms.

Keywords: linear mixed-effects model; interactions; missing data; censored data; EM algorithm; SEM algorithm



Citation: Zakkour, A.; Perret, C.; Slaoui, Y. Stochastic Expectation Maximization Algorithm for Linear Mixed-Effects Model with Interactions in the Presence of Incomplete Data. *Entropy* **2023**, *25*, 473. <https://doi.org/10.3390/e25030473>

Academic Editors: Sergio Curilef and Francisco Calderón

Received: 9 February 2023

Revised: 27 February 2023

Accepted: 2 March 2023

Published: 8 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The time between the presentation of a stimulus and a participant's motor response is the oldest and most widely used measure for exploring the functioning of the human mind. In 1869, ref. [1] theorized this duration, called reaction time (RT), as involving three sets of activities: perceptual mechanisms, cognitive processing and motor preparation. Based on the assumption that the first and last sets of processing can be considered as having virtually identical duration for the same task, any change in RTs between two experimental conditions is then interpreted as indicating a change in the duration of cognitive processing. RT is then considered by psychologists as a tool to explore cognitive processing mechanisms ([2]).

Psycholinguistics research on the cognitive mechanisms involved in language recognition or production frequently uses RT as a measure of behavior. Like all scientific disciplines, psycholinguistics relies on hypothesis testing to support its theoretical propositions. Since the early 2000s, researchers have taken up linear mixed-effects models (LMEMs). As described by [3], a LMEM allows for the proper consideration of one of the characteristics of psycholinguistic experiments: the presence of two random-effect variables. Indeed, the experimental structure of the experiments conducted in this field involves having a group of participants process a set of stimuli (see hereafter). Thus, the statistical analysis must allow the inclusion of these two random-effect variables, i.e., participants and items, in the structure of the model. The introduction of LMEM was thus an important methodological advance for psycholinguistics.

There is, however, one point that has not received much attention from psycholinguists. Experimentation with human beings is often subject to many vagaries. Imagine researchers whose goal is to understand how an adult retrieves the spelling of a word from their

memory ([4]). They ask a group of thirty participants to write down the names of 150 images presented on a computer screen. A device allows them to measure the time between the appearance of the image on the screen and the first writing gesture of the participant. These researchers can potentially collect 4500 RT values. However, between trials lost for technical reasons, spelling errors, and certain habits during data analysis (e.g., censoring of data greater than two standard deviations from the mean, right-censoring data), a non-negligible number of data are removed. For example, ref. [4] reported removing just over 20% of the data.

The issue of missing and censored data has received relatively little attention in psycholinguistics (see, however, [5,6]). This is especially true since the introduction of LMEMs. Psycholinguists assume that these models can be run on a sample of data with “holes”. The analysis strategy is then of the “keep the case empty” type, ignoring the bias that this introduces in the estimation of the parameters of the model and thus of the decisions taken. The objective of this work is thus to develop algorithms to manage the presence of missing and censored data in these psycholinguistic experiments. Two points are to be taken into account. On one hand, experimental designs in scientific psychology may involve assumptions about interactions between two or more fixed-effect variables. On the other hand, researchers suggest, for theoretical reasons, that all interactions between fixed-effect variables and random-effect variables made possible by the model should be included ([7]). The potential presence of these two types of interactions (fixed–fixed variables; random–fixed variables) are constraints on the development of the missing data procedure.

Let us first recall that the LMEM ([8]) is an extension of the simple linear model that allows both fixed and random effects to be represented. In 1861, the LMEM was introduced under the name of a one-way random-effects model ([9]), that is, a model with one random variable and without any fixed variable. From 1990 and onward, ref. [9] underlined that LMEMs became popular in many research applications including economics, sociology, insurance, agronomy, epidemiology, genetics, etc. They are used in longitudinal data analysis, multilevel modeling and small estimations. The analysis of this type of models is presented in [10]; for more literature about the methodology, theoretical results and software, see the books [11–14]. LMEMs are fitted and analyzed in R by using the package `lme4` or `lmerTest` ([15]).

In [16], the author presented the fixed interactions between two factors in an LMEM, using the maximum product interaction F-test. Ref. [17] was interested in the sample size of an LMEM. He proposed a formula to estimate the sample size based on testing a mixed model that contained fixed interactions. In 2019, ref. [18] proposed an estimation method to recover the principal effects and interactions, because the existing method did not allow the integration of these effects in a mixed data frame. They approved that the proposed method gave optimal results to their applications’ conditions. On the same side, ref. [19] presented the estimation of the fixed interactions. These interactions are normally introduced by the product of the variables, but the algebraic transformations revealed that this technique did not produce a within-unit estimator. A Monte Carlo method confirmed that the FE (fixed-effects) estimator of (x, z) was biased, if one of these variables was correlated with another one. In order to present the interaction between x and z , it is possible to use the current syntax $x * z$. This consideration is called “double-demeaned”; it is less efficient than the standard FE and only works with $T < 2$. For the application in R, the Markov chain Monte Carlo method is applied using the package `mcmc`. Ref. [20] introduced the `MCMCpack` package that contains functions to perform Bayesian inference using a posterior simulation for a number of statistical models (<https://CRAN.R-project.org/package=MCMCpack> (accessed on 14 Jun 2011)), while [21] presented the package `MCMCglmm` (<https://cran.r-project.org/web/packages/MCMCglmm/index.html> (accessed on 2 February 2010)) for the MCMC method to fit the generalized linear mixed-models for multiple response vectors. This method was also used by [22] to identify unknown parameters in the biological field, such as detecting the concentration of target molecules, because of the importance of this

method for extracting the information. The MCMC method was applied in the work of [23] to solve mechanical problems by using a Bayesian method.

Ref. [24] presented five types of LMEMs by introducing two random variables (see Appendix A.1), where the models were compared to show the importance of including the maximum effects that could affect the model.

One of the solutions that handles the missing data can be the expectation–maximization (EM) algorithm. It is a very useful algorithm for the estimation of the maximum likelihood function. This method can be a solution when the only data available do not allow the estimation of the parameters ([25]), and/or the expression of the likelihood is analytically impossible to maximize.

In other words, it aims to provide an estimator when the problem comes from the presence of missing data. When the data are incomplete, the knowledge of these values would make it possible to estimate the parameters. The EM algorithm takes its name from the fact that at each iteration, it operates two distinct steps:

Step E: This step is called the expectation (E) step, where we are interested in finding the expected value of the unobserved or unknown variables given the observed data and the value of the parameters.

Step M: This step is called the maximization (M) step; in this step, we maximize the expected log-likelihood by using the estimation of the unknown data carried out in the previous step. These parameter estimates are then used to determine the distribution of the unknown variables in the next iteration.

At some points, the expectation or the maximization steps are impossible to apply directly ([26]); from there, the use of an extension form of EM is useful, such as MCEM, where the (E) step is replaced by a Monte Carlo simulation, or SAEM, where the (E) step is replaced by a stochastic approximation. SAEM was a solution of nonlinear mixed-effects models (NLMEM). Ref. [27] proposed a new methodology for maximum likelihood estimation in mixtures of nonlinear mixed-effects models. The resulting MSAEM (mixture SAEM) algorithm is now implemented in the *MonoLix* software tool.

The aim of this paper was to perform the SEM algorithm, under an LMEM by including two types of incomplete data (the censored and the MAR types) and by taking into consideration for the first time the interactions, where our proposed model contains the interactions between fixed variables and fixed–random variables. This document is organized as follows: In Section 2, we present three algorithms based on the expectation–maximization (EM) method, the first one is called the SAEM algorithm, the second is our proposed SEM algorithm and the third one is based on the MCMC method. We also present the incomplete data types, divided into missing data and censored one. In Section 3, we define the proposed model with some specific cases. In Section 4, we introduce a method to achieve the convergence. In Section 5, we compare the results obtained from simulated psychological and real data. In Section 6, we conclude the proposed study with some perspectives and future directions.

2. Methodology

2.1. EM Algorithm

As previously said, the EM algorithm is a widely used algorithm in the case of incomplete data; in this situation, the maximum likelihood function is difficult or impossible to use to estimate the parameter vector θ of the considered model. We formalize directly an iteration from which we can understand clearly how this algorithm works:

- Let $y = (y_1, \dots, y_n)$ be the independent and identically distributed (i.i.d.) observations of likelihood $p(y|\theta)$.
- The maximization of $\log p(y|\theta)$ is impossible.
- We consider hidden data $z = (z_1, \dots, z_n)$ which make the maximization of the likelihood of the complete data possible when known.

- As we do not know these data z , we estimate the likelihood of the complete data by taking into account all the known information so the estimator is given as follows (E step):

$$Q(\theta|\theta_{k-1}) = \mathbb{E}[\log p(y, z; \theta|y; \theta_{k-1})], \quad (1)$$

where θ_{k-1} is the vector of the parameters at iteration $k - 1$.

- Finally, we maximize this estimated likelihood to determine the new value of the parameter (M step). Thus, the transition from iteration $k - 1$ to iteration k in the algorithm consists in determining the parameters vector at iteration k , θ_k :

$$\theta_k = \arg \max_{\theta} Q(\theta|\theta_{k-1}).$$

where θ_0 is chosen arbitrarily. When one of these two steps are impossible to complete, we can consider an extension form of the EM algorithm such as the SAEM, SEM or MCEM algorithms. In the next subsection, we derive the SAEM algorithm into the formula of the EM algorithm.

2.1.1. SAEM Algorithm

The stochastic approximation expectation maximization (SAEM) algorithm was proposed by [26], in which the E step was replaced by a stochastic approximation. The stochastic approximation algorithm was first introduced by [28] and also used by [29], where the algorithm generates iterates of the form:

$$\theta_k = \theta_{k-1} - \gamma_k [h(\theta_{k-1}) + \epsilon_k],$$

where γ_k is a sequence of positive step sizes, h is a function of θ , and ϵ_k is a constant such that $\epsilon_k = \mathbb{E}[h(\theta_k)]$. From this form, the SAEM algorithm is obtained where the E step of the EM algorithm is divided into two steps; consider the iteration k :

First, we sample a realization z_k of the latent variable from the conditional distribution ($p_{\theta_{k-1}}(z|y)$) of z given y , using the value of the parameter θ_{k-1} at iteration $k - 1$.

Second, by using the realization z_k from the first step, we update the value of $Q_k(\theta|\theta_{k-1})$ (see, (1)) through a stochastic approximation procedure.

Then, the algorithm continues as follows:

Initialization step: Initialize θ_0 in a fixed compact set.

Then, for all $k \geq 1$, the k th iteration consists in three steps:

Simulation step: simulate z_k from the conditional distribution $p_{\theta_{k-1}}(\cdot|y)$.

Stochastic approximation step: compute the quantity

$$Q_k(\theta) = Q_{k-1}(\theta) + \gamma_k \left(\frac{1}{m(k)} \sum_{j=1}^{m(k)} \log f(z_k(j); \theta) - Q_{k-1}(\theta) \right),$$

where m_k is the number of simulations at each iteration.

Maximization step: update the parameter value according to $\theta_k = \arg \max_{\theta} Q_k(\theta)$.

The SAEM algorithm is more efficient compared to the MCEM, where, at each iteration, the simulation of the missing values is repeated and the ones obtained previously are not used. In the SAEM, all the simulated missing values contribute to the expectation step and that is the advantage of this algorithm where the maximization step is cheaper than the simulation one. Next, we consider the SEM algorithm to show how it differs from an SAEM.

2.1.2. SEM Algorithm

The SEM approach is a specific case of the SAEM algorithm. In the SEM algorithm, the step size is set to zero $\gamma_k = 0$ and the number of simulations by iteration is constant (usually, $m_k = 1$).

This method is an extension of the EM algorithm and is a solution when the maximum likelihood is impossible to complete. This algorithm was introduced by [30], where between the steps E and M, they used a stochastic step that consisted in generating a complete sample containing latent variables from the conditional density, based on the observed data. Starting with an initial position θ_0 , an iteration k of the SEM algorithm, where, to each θ_k , a θ_{k+1} is associated, is defined as follows:

Step E: Compute the conditional density $f(y|x; \theta_k)$;

Step S: Draw z_k from the conditional distribution, then obtain the complete sample $y_k = (x, z_k)$;

Step M: Update the parameters θ_{k+1} by maximizing the likelihood function based on the complete vector y_k .

In the next subsection, we consider the MCEM algorithm to show how a Monte Carlo simulation is used in the EM algorithm.

2.1.3. MCEM Algorithm

In 1990, ref. [31] proposed to replace the expectation step to compute $Q(\theta|\theta_{k-1})$ by a Monte Carlo integration, hence the name MCEM algorithm (Monte Carlo EM). At iteration k , the E step is replaced by the following procedure:

Simulation step (S-step): generate $m(k)$ realizations $z_k(j)$ (with $j = 1, \dots, m(k)$) of the missing data vector under the distribution function $p(z; \theta_k)$

Monte Carlo integration: compute the approximation of $Q(\theta|\theta_{k-1})$ according to:

$$Q_k(\theta) = \frac{1}{m(k)} \sum_{j=1}^{m(k)} \log f(z_k(j); \theta).$$

The **maximization step** remains unchanged.

As said before, these algorithms can be used in such a way as to handle the unobserved values of the data that are defined in this next subsection.

2.2. Incomplete Data

We can talk about incomplete data when the values in our response vector are not all observed for many reasons; we can also say it is a question without an answer.

These types of data are a common problem recognized by statisticians. Ref. [32] presented many solutions to handle this problem based on simple and multiple imputations. Ref. [33] proposed to handle the incomplete data in a hierarchical model by using the SEM algorithm.

In this work, we are interested in data that contain two types of incomplete values: censored data denoted by Y_c and missed data Y_{miss} . In the next subsection, the censored data are considered.

2.2.1. Censoring

Censored data are any data for which we do not know the exact event time.

There are two types of censored data: right-censored and left-censored.

Right-censored data: these are data where the event has not yet been achieved when the study is finished.

Left-censored data: these are data when the event is achieved before the study starts.

We consider in this paper both censoring types of (left- and right-censoring) data with four percentages (0%, 5%, 10% and 20%). Let t be the censoring level and T the set of censored indices that can be written as $T = \{(i_1, \dots, i_{p+k}) \in I, Y_{i_1, \dots, i_{p+k}, j} < t \text{ or } Y_{i_1, \dots, i_{p+k}, j} >$

$t\}$, and let $Y = (Y_o, Y_c)$, with $Y_o = \{Y_{i_1, \dots, i_{p+k}, j}, (i_1, \dots, i_{p+k}) \notin T\}$ the vector of observed response variables and $Y_c = \{Y_{i_1, \dots, i_{p+k}, j}, (i_1, \dots, i_{p+k}) \in T\}$ defining the vector of censored response variables. The second type is the missing data, defined in the next subsection.

2.2.2. Missing Data

In general, missing values are produced when a value in the data is not represented for a given variable, for many reasons that can be linked to the objective of the study (for example, participants do not answer the questions). This appears in many research studies, particularly when collecting data and where participants are studied over a period of time.

At first, studies were developed assuming no missing values. In the late 1980s, with the advancement in technology, that problem attracted the attention of many researchers who wished to study several techniques for handling it. Depending on the reasons for their absence, these values could be divided into three types: missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR).

In Appendix A.2, we define the types of missing value and the methods to generate these types in a simulation study.

In this paper, we are interested in imputing the MAR type of missing data with four percentages (0%, 5%, 10% and 20%), which are crossed with four other percentages of censoring, resulting in 16 cases of incomplete data. For the missing data, we denote by Y_o the vector of observed response variable and Y_{miss} the missed one, so $Y = (Y_o, Y_{miss})$. In the next section, we present the main results of this work.

3. Main Results

3.1. The Proposed Model

Motivated by psychological data, in which we typically have two random variables, participants designed by S and items designed by I , we consider a model that contains p covariates $\{x_1; x_2; \dots; x_p\}$ and two random variables. The predicted variable, also named the variable of interest, is denoted by Y . Then, we consider the following linear mixed model:

$$Y = \beta_{0,s,i} + \beta_{1,s,i}x_1 + \beta_{2,s,i}x_2 + \dots + \beta_{p,s,i}x_p + r_{j,i,s}, \quad (2)$$

with r the residual of the model that follows a normal distribution. We give:

$$\begin{cases} \beta_{0,s,i} = \delta_0 + S_{0,s} + I_{0,i} \\ \beta_{1,s,i} = \delta_1 + S_{1,s} + I_{1,i} \\ \beta_{2,s,i} = \delta_2 + S_{2,s} + I_{2,i} \\ \vdots \\ \beta_{p,s,i} = \delta_p + S_{p,s} + I_{p,i} \end{cases}$$

By a replacement of these considerations in Equation (2), our model can be rewritten as follows:

$$\begin{aligned} Y &= \delta_0 + \delta_1x_1 + \delta_2x_2 + \dots + \delta_px_p \\ &+ S_{0,s} + S_{1,s}x_1 + S_{2,s}x_2 + \dots + S_{p,s}x_p \\ &+ I_{0,i} + I_{1,i}x_1 + I_{2,i}x_2 + \dots + I_{p,i}x_p \\ &+ r. \end{aligned}$$

By taking

$$X = [1 \quad x_1 \quad x_2 \quad \dots \quad x_{p+k}] = [1 \quad x_j], \delta = (\delta_0 \quad \delta_j)^T$$

and denoting by T the transpose matrix,

$$Z = [z_j] = [x_j], u = [u_j] = [S_{j,l,s} + I_{j,l,i}]^T;$$

and $e_{jl} = [e_j]$.

This leads us to rewrite our model as follows:

$$Y = X\delta + \sum_{k=1}^K z_k u_k + e = X\delta + Zu + e,$$

where $\mathbb{E}(Y) = X\delta$ and $\text{Var}(Y) = \mathbb{V} = ZGZ^T + R$, with $u \sim \mathcal{N}(0, G)$, $e \sim \mathcal{N}(0, R)$ and $\text{Cov}(u, e^T) = 0$.

Now, by including all the interactions, the general model can be written as:

$$Y = F + I_{FF} + R + I_{FR} + e,$$

where we take:

$$\begin{aligned} F &= \underbrace{\delta_0 + \delta_1 x_1 + \dots + \delta_p x_p}_{\text{fixed variables}}, \\ I_{FF} &= \underbrace{\delta_{p+1} x_{p+1} + \dots + \delta_{p+k} x_{p+k}}_{\text{interaction between fixed-fixed variables}}, \\ R &= \underbrace{(S_{1,s} x_1 + I_{1,i} x_1) + \dots + (S_{p,s} x_p + I_{p,i} x_p)}_{\text{random variables}}, \\ I_{FR} &= \underbrace{(S_{p+1,s} x_{p+1} + I_{p+1,i} x_{p+1}) + \dots + (S_{p+k,s} x_{p+k} + I_{p+k,i} x_{p+k})}_{\text{interaction between fixed-random variables}}, \\ \text{and } e &= \underbrace{(S_{0,s} + I_{0,i}) + r}_{\text{residuals}}. \end{aligned}$$

We set k , the number of interactions, as $k = 2^p - p - 1$ and let $j, l \in [1, p + k]$.

Our random participants variable is given as follows:

$$S_{j,l,s} \sim \mathcal{N}(0, \Sigma_S),$$

and the variance–covariance matrix of the participants is written as

$$\Sigma_S = \begin{cases} \sigma_{j,s}^2 & ; \text{if } j = l \\ \rho_{s,s} \sigma_{j,s} \sigma_{l,s} & ; \text{otherwise.} \end{cases}$$

We consider that the items random variable also follows a normal distribution:

$$I_{j,l,I} \sim \mathcal{N}(0, \Sigma_I);$$

then, the variance–covariance matrix of the items is given by:

$$\Sigma_I = \begin{cases} \sigma_{j,I}^2 & ; \text{if } j = l \\ \rho_{I,I} \sigma_{j,I} \sigma_{l,I} & ; \text{otherwise.} \end{cases}$$

For $j, l \in [1, p + k]$, the variance–covariance matrix G is equal to:

$$G = \Sigma_u = \begin{cases} \sigma_{j,s}^2 + \sigma_{j,I}^2 & ; \text{if } j = l \\ \rho_{s,s} \sigma_{j,s} \sigma_{l,s} + \rho_{I,I} \sigma_{j,I} \sigma_{l,I} & ; \text{otherwise.} \end{cases}$$

The goal is to predict our model by determining the parameters δ and u ; we propose to use Henderson's linear system ([34]) based on the maximum likelihood function. This approach is presented in Appendix A.3. The matrix R has the form:

$$R = \begin{cases} I & ; \text{if } j = l \leq n - p - k \\ \sigma_{e_j}^2 & ; \text{if } j = l > n - p - k \\ 0 & ; \text{otherwise,} \end{cases}$$

where n represents the total number of observations. Therefore, with $Z = (z_1, \dots, z_{p+k})$, the variance of Y is equal to:

$$\begin{aligned} \mathbb{V} = & \left[\left(\sum_{j=1}^{p+k} z_j \Sigma_{u_1, u_j} \right) z_1 + \left(\sum_{j=1}^{p+k} z_j \Sigma_{u_2, u_j} \right) z_2 + \dots + \left(\sum_{j=1}^{p+k} z_j \Sigma_{u_p, u_j} \right) z_p + \dots \right. \\ & \left. + \left(\sum_{j=1}^{p+k} z_j \right) \Sigma_{u_{p+k}, u_j} z_{p+k} \right] + R. \end{aligned}$$

The linear system equations of Henderson applied in this study is also presented in Appendix A.4. In the next subsection, we present some specific cases.

3.2. Specific Cases

In this section, we show how the model can be simplified if we take the following two cases: first, only the fixed–fixed interaction part is presented and second, there is no interaction.

3.2.1. Case 1: Fixed–Fixed Interaction

In this case, we consider the interactions between the fixed variables. Then, in the random part, the u vector is equal to 0:

$$\begin{aligned} u &= [u_1 \quad u_2 \dots u_p \quad u_{p+1} \dots u_{p+k}]^T \\ &= [0 \quad 0 \quad \dots \quad 0 \quad 0 \quad \dots \quad 0]^T; \end{aligned}$$

therefore, the \mathbb{V} matrix is equal to:

$$\mathbb{V} = R = \begin{cases} I & ; \text{if } j = l \leq n - p - k \\ \sigma_{e_j}^2 & ; \text{if } j = l > n - p - k \\ 0 & ; \text{otherwise.} \end{cases}$$

For more mathematical development, see Appendix A.4.

3.2.2. Case 2: No Interactions

In this case, there are no interactions between the variables; our model is simplified, where the random part with the fixed–fixed interaction is ignored:

$$X = [1 \quad x_j]; \delta = (\delta_0 \quad \delta_j)^T; Z = [z_j]; u = [u_j] = [0 \quad 0 \quad \dots \quad 0]^T; e_{jl} = [e_j].$$

The \mathbb{V} matrix is reduced to:

$$\mathbb{V} = R = \begin{cases} I & ; \text{if } j = l \leq n - p \\ \sigma_{e_j}^2 & ; \text{if } j = l > n - p \\ 0 & ; \text{otherwise.} \end{cases}$$

See the simplified system in Appendix A.4. Next, we present how the SEM algorithm can be applied in this work.

3.3. SEM Algorithm

In this paper, we propose to handle missing data and censored problems in the presence of interactions by using the SEM algorithm, proposed by [33]. The stochastic expectation maximization (SEM) algorithm is a method used to estimate the parameters when it is complicated to use the EM algorithm; it is a particular case of multiple imputations (MI).

To understand MI ([35]), we need to know the idea behind a simple imputation (SI) in which the nonobserved data $\{Y_{no}\}$ (missed or censored) are replaced by a value and then the parameters are estimated using known methods to maximize the likelihood function. To obtain a robust estimate, the simple imputation can be repeated several times with different values of $\{Y_{no}\}$ and the results can be combined. This method is called multiple imputations.

We applied the algorithm after crossing the MAR values with the censored ones (see Algorithm 1). In the SEM algorithm ([30]), the values of $\{Y_{no}\}$ are drawn from the conditional distribution of the nonobserved data given the observed ones using the current values of the parameters. We generated samples from $P_{\Theta}(Y_{no}|Y_o)$ where this distribution was calculated using Gibbs's sampling; the procedure is underlined in Algorithm 1 (the SEM method using Gibbs's sampling was developed in [33]). For more information about the expectation maximization algorithm, see [36]. In the next subsection, we present the extension form of the SEM that leads us to the SAEM algorithm.

Algorithm 1 SEM algorithm: N is the number of iterations of the SEM algorithm, M is the burn-in level, Y_1, \dots, Y_n is the response vector, G is the number of iterations of Gibbs sampling, $Y[u_i]$ is the response vector with respect to the i th random variable, and fixed is the summation of the fixed effects with the fixed interaction part.

Input: N, M, Y_1, \dots, Y_n , and G .

```

1: Random initialization of  $\Theta$ 
2: for  $j = 1, \dots, N$  do
3:   for  $g = 1, \dots, G$  do
4:     draw  $e^{(g)}$  from  $\mathcal{N}(0, \sigma_e^2, \text{lower} = \min(e) - \text{fixed} - u^{(g-1)}, \text{upper} = \max(e) - \text{fixed} - u^{(g-1)})$ 
5:     draw  $u_1^{(g)}$  from  $\mathcal{N}(0, \Sigma_{u_1}, \text{lower} = \max(\min(Y[u_1]) - u_2^{(g-1)} - e^{(g)}), \text{upper} = \min(\max(Y[u_1]) - u_2^{(g-1)} - e^{(g)}))$ 
6:     draw  $u_2^{(g)}$  from  $\mathcal{N}(0, \Sigma_{u_2}, \text{lower} = \max(\min(Y[u_2]) - u_1^{(g)} - e^{(g)}), \text{upper} = \min(\max(Y[u_2]) - u_1^{(g)} - e^{(g)}))$ 
7:   end for
output:  $Y_{no}$  obtained from sampled  $(u^{(G)}, e^{(G)})$ 
8:    $\Theta_{j+1} = \arg\max_{\Theta} L(\Theta|Y_o, Y_{no})$ 
9: end for
10:  $\hat{\Theta} = (N - M)^{-1} \sum_{j=M+1}^N \Theta_j$ 
output:  $\hat{\Theta}$ 

```

3.4. SAEM Algorithm

In the SEM algorithm, Θ_{j+1} (the parameter at state $j + 1$) depends only on Θ_j and Y_j . By taking T the operator of the EM algorithm and M , the operator which associates Θ_{j+1} with Y_j via step M, the updated parameter Θ_{j+1} can be written as follows (see [30]):

$$\Theta_{j+1} = T(\Theta_j) + V(\Theta_j, Y_j),$$

where $V(\Theta_j, Y_j) = M(Y_j) - T(\Theta_j)$.

The SAEM algorithm (see also Algorithm 2) is an extension form of the SEM algorithm. Starting with an initial position Θ_0 , the iteration of the SAEM algorithm, to which for each Θ_j , a Θ_{j+1} is associated, is defined by the equation:

$$\Theta_{j+1} = T(\Theta_j) + \gamma_j V(\Theta_j, Y_j),$$

where γ_j is the step size, which is a decreasing sequence of positive real numbers starting with $\gamma_0 = 1$, $\gamma_j \mapsto 0$ when $j \mapsto \infty$. The theoretical and practical performance values of the SAEM algorithm are strongly dependent on the speed of convergence towards 0 and the regular decreasing of the sequence γ_j , which explains the importance of choosing the step size.

Ref. [37] performed many tests to see which sequence was more efficient, and they obtained two types of sequences that were significant, one of a slow mode:

$$\gamma_j = \cos\left(\frac{j \times \pi}{2N}\right),$$

the other type was of a linear mode:

$$\gamma_j = -\frac{(j+1)}{N}.$$

Ref. [38] chose the decreasing sequence from their previous experiments [39]; they took that sequence as

$$\gamma_j = \cos(N \times \alpha) \mathbb{1}_{\{0 \leq j \leq 20\}} + \frac{c}{\sqrt{j}} \mathbb{1}_{\{j > 20\}}, \text{ where } \cos(20 \times \alpha) = \frac{c}{\sqrt{20}} = 0.3.$$

That sequence was also chosen by [40,41] in a nonlinear mixed model as 1 at the first iteration and $(j - N)^{-1}$ during the last iterations.

Algorithm 2 SAEM algorithm: N is the number of iterations of the SEM algorithm, M is the burn-in level, Y_1, \dots, Y_n is the response vector, G is the number of iterations of Gibbs's sampling, γ_j is the decreasing sequence, $Y[u_i]$ is the response vector with respect to the i th random variable, and fixed is the summation of the fixed effects with the fixed interaction part.

Input: N, M, Y_1, \dots, Y_n , and G .

```

1: Random initialization of  $\Theta$ 
2: for  $j = 1, \dots, N$  do
3:   for  $g = 1, \dots, G$  do
4:     draw  $e^{(g)}$  from  $\mathcal{N}(0, \sigma_e^2, \text{lower} = \min(e) - \text{fixed} - u^{(g-1)}, \text{upper} = \max(e) - \text{fixed} - u^{(g-1)})$ 
5:     draw  $u_1^{(g)}$  from  $\mathcal{N}(0, \Sigma_{u_1}, \text{lower} = \max(\min(Y[u_1]) - u_2^{(g-1)} - e^{(g)}), \text{upper} = \min(\max(Y[u_1]) - u_2^{(g-1)} - e^{(g)}))$ 
6:     draw  $u_2^{(g)}$  from  $\mathcal{N}(0, \Sigma_{u_2}, \text{lower} = \max(\min(Y[u_2]) - u_1^{(g)} - e^{(g)}), \text{upper} = \min(\max(Y[u_2]) - u_1^{(g)} - e^{(g)}))$ 
7:   end for
output:  $Y_{no}$  obtained from sampled  $(u^{(G)}, e^{(G)})$ 
8:    $\Theta_{j+1} = \operatorname{argmax}_{\Theta} \gamma_j \times L(\Theta | Y_o, Y_{no})$ 
9: end for
10:  $\hat{\Theta} = (N - M)^{-1} \sum_{j=M+1}^N \Theta_j$ 
output:  $\hat{\Theta}$ 

```

When applying these two algorithms, we were confronted with a problem of convergence caused by the presence of the interaction between the fixed and random variables.

Therefore, in the next section, we present how to handle this problem by using the Hamiltonian Monte Carlo algorithm.

4. Convergence of Parameters

While introducing the fixed-random effect in the application section, we faced a convergence problem in the SEM and SAEM algorithms when using the standard form of Monte Carlo (MC). In order to solve this problem, we propose to use the hybrid MC algorithm ([42]), which uses the symmetric Metropolis–Hastings algorithm to accept or reject a proposal based on the Hamiltonian function, hence the name of the algorithm, the Hamiltonian Monte Carlo algorithm (HMC). In the next subsection, we present how to implement the HMC algorithm.

Implementation

The Hamiltonian function $H(p, q)$ is written in term of the joint density $\pi(p, q)$:

$$H(p, q) = -\log \pi(p, q),$$

which decomposes into two terms:

$$\begin{aligned} H(p, q) &= -\log \pi(p|q) - \log \pi(q) \\ &= K(p, q) + V(p, q). \end{aligned}$$

The first term corresponds to the density of the target distribution, the second one is determined by the target distribution when $K(p, q)$ is unconstrained and must be specified by the implementation.

The acceptance probability of moving from state (p_0, q_0) to (p_i, q_i) is determined using the Metropolis–Hastings algorithm ([43,44]), with $Q(p_i, q_i|p_0, q_0)$ being the probability density defining each proposal:

$$a(p_i, q_i|p_0, q_0) = \min \left(1, \frac{Q(p_0, q_0|p_i, q_i)\pi(p_i, q_i)}{Q(p_i, q_i|p_0, q_0)\pi(p_0, q_0)} \right).$$

Referring to the symmetric Metropolis ([45]), the acceptance probability is simplified to the form:

$$\begin{aligned} a(p_i, q_i|p_0, q_0) &= \min \left(1, \frac{\pi(p_i, q_i)}{\pi(p_0, q_0)} \right) \\ &= \min \left(1, \frac{\exp(-H(p_i, q_i))}{\exp(-H(p_0, q_0))} \right) \\ &= \min(1, \exp(-H(p_i, q_i) + H(p_0, q_0))). \end{aligned}$$

In this study, we considered that we had evidence of the convergence towards stationarity due to the Markov Chain (see, [45,46]).

In the next section, we present some numerical applied results based on a psychological simulation and then real data.

5. Numerical Experiment

This section aims to compare the SEM algorithm proposed in this paper with other methods in the presence of incomplete data in a complete LMEM. The method was first applied to simulated data, then to real ones.

5.1. Simulated Data

The simulation data were created from an experimentally obtained database. In order to explore the cognitive process of retrieval in memory of the spelling of French words, 30 participants had to handwrite the label of 150 drawings of objects, constituting a database

of 4500 RTs ([4]). After removing the RTs corresponding to technical errors, spelling errors, and censoring data greater than 2.5 standard deviations from the mean, 3434 values remained. Ref. [4] performed a linear regression analysis involving nine fixed-effects variables. In the present work, we retained two of them. The first was an estimate of the age at which the image name was acquired in childhood (age of acquisition, AoA hereafter). This factor is one of the most important predictors of picture-naming RTs ([47]). The second fixed-effects variable corresponded to the number of letters in a picture label (Lett_cat). We performed a median split to obtain a categorical variable. Thus, image labels with a number of letters lower than the sample average were considered as short words and those with a higher number of letters were considered as long words. The interaction between these two fixed-effects variables was included in the model. Finally, there were two random effect variables: items, i.e., the 150 image labels produced by the 30 participants, and the participants themselves. The fixed-random interaction between AoA and the participants was also included in the LMEM.

Therefore, the model could be written as follows:

$$\text{RTs} = b_0 + b_1 \times \text{AoA} + b_2 \times \text{Lett_cat} + b_3 \times \text{AoA} \times \text{Lett_cat} \\ + \text{participants} + \text{items} + \text{AoA} \times \text{participants} + e.$$

In this section, we illustrate the proposed algorithm by comparing the results to the SAEM and MCMC algorithms using different proportions of incomplete data. The comparison was performed by computing the mean absolute error (MAE), which measured the errors between the reference and the predicted vectors (with Θ as the reference value and $\hat{\Theta}$ as the predicted one); the standard equation of the MAE is given by:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{\Theta}_i - \Theta_i|.$$

We also computed the linear correlation between the parameters $\left(\text{LCor} = \frac{\text{Cov}(\hat{\Theta}_i, \Theta_i)}{\sigma(\hat{\Theta}_i)\sigma(\Theta_i)} \right)$ and the Spearman rank correlation $\left(\text{RCor} = 1 - \frac{6 \sum d^2}{n(n^2-1)} \right)$, where d^2 is the square of the difference in the ranks of the two coordinates).

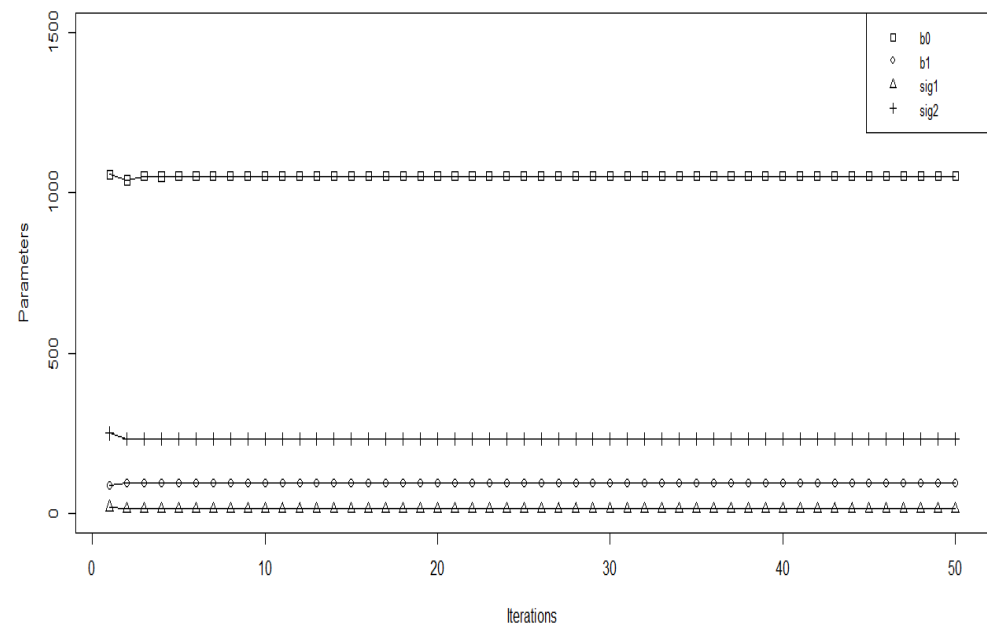
In this numerical section, we provide the 16 cases to see the performance of the methods in the absence of one of these two considered types of unobserved values (MAR type/censored data) or in the presence of a low or high percentage. In addition, we used the simulated data of 4500 observations after testing our proposed methods on a larger number of observations. We obtained the same results as those for 4500 values.

After the implementation of our algorithm, we checked the convergence by illustrating the values of the estimated parameters obtained at each iteration; this is presented in Figure 1.

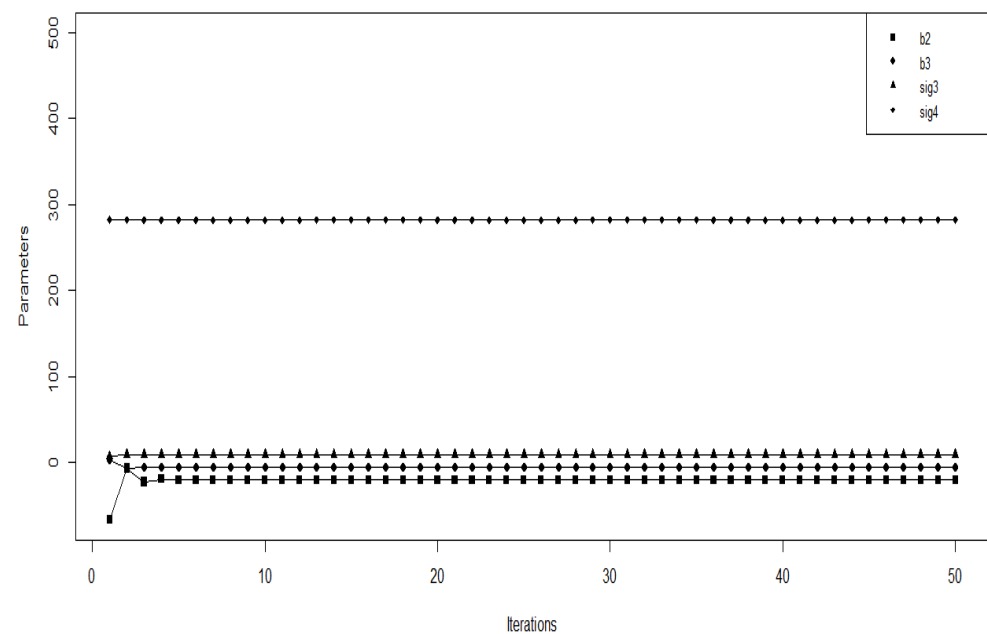
In Figure 2, we plotted the normalized residuals of each case. We observed a similar dispersion between the standard simulated data and the SEM algorithm, while we noticed a small but visible scattering by implementing the SAEM algorithm, and finally, we observed that when we used the MCMC method, the residuals were dispersed far from the normal line.

Our results are visualized quantitatively in Table 1. Based on the MAE value, we can observe that the SEM gave the lowest values in all cases except for the case of 5% MAR and 20% censoring, where the SAEM (8.108) beat the SEM (8.118) by a very narrow margin. By observing the full vectors, the SEM gave the best parameter values in two cases: (0%, 10%) and (20%, 5%). In the other cases, we observed that with a high percentage of MAR (20%), b_2 and b_3 in the SAEM method gave the closest values while for the other parameters, it was the SEM algorithm. In all cases, we see that the MCMC was not a good choice either for the parameters values or for the other measurements. Moreover, we can see that for LCor (respectively, Rcor), the SEM and the SAEM algorithms gave approximately the same results 0.999 (respectively, 1) in all cases. This showed that these two methods had a small

difference except for the case of (20%, 5%) where the RCor of the SAEM algorithm decreased from 1 to 0.9285. The MCMC had an RCor ranging between a maximum value of 1 and a minimum value of 0.738; that maximum value was obtained with 5% of MAR and the absence of censored data.



(a)



(b)

Figure 1. Value of the parameter estimates at each the SEM iteration with a burn-in period $M = 10$, we separate the parameters according to their values in (a), we plot b_0 , b_1 , sig_1 and sig_2 and in (b), we plot b_2 , b_3 , sig_3 and sig_4 .

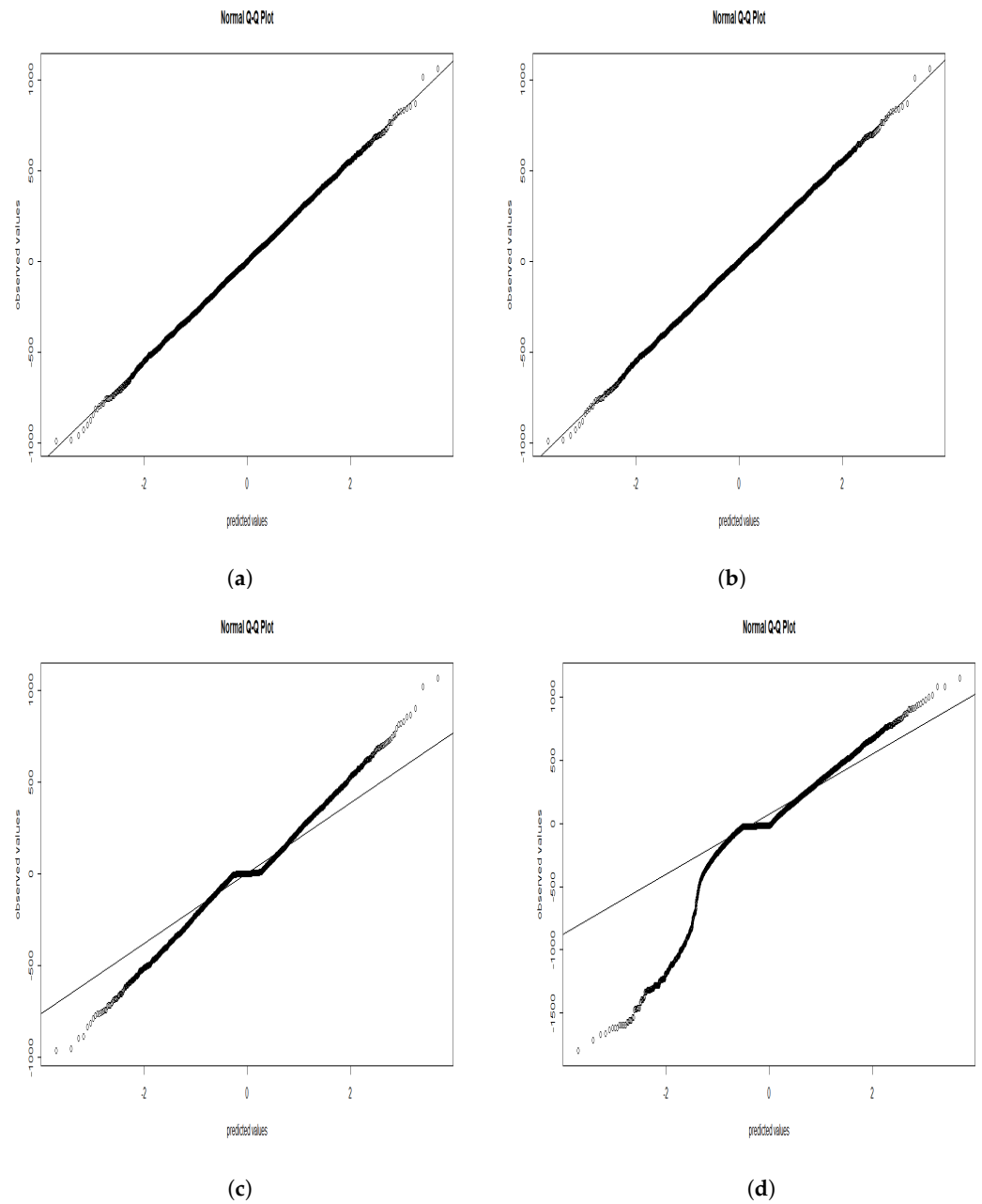


Figure 2. Normalized residual presented; (a)-standard simulated data, (b)-SEM algorithm, (c)-SAEM algorithm, (d)-MCMC algorithm.

Table 1. Comparison of the SEM algorithm with the SAEM and MCMC algorithms in the complete model with the different percentages of incomplete data where the number of iterations was 50. sig₁, sig₂, sig₃ and sig₄ represent the standard deviations of the participants, items, the I_{FR} part and the error, respectively. The percentages on the left present the MAR, and on the right are the percentages of censored data.

	(0%, 0%)			(0%, 5%)			(0%, 10%)			(0%, 20%)		
$\hat{\Theta}_i$	Reference			SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC
b ₀	1045			1044	1045	1068	1044	1043	1010	1047	1047	822.8
b ₁	93.16			93.54	88.51	48.82	93.91	89.37	41.25	93.51	88.84	77.96
b ₂	−52.28			−51.30	−42.33	−36.75	−52.50	−47.30	51.49	−30.27	−51.66	18.26
b ₃	2.80			3.17	−0.18	6.92	4.43	5.52	−14.44	−1.79	5.02	−18.20
sig ₁	14.57			14.65	14.43	5.29	14.65	12.54	12.56	14.70	12.17	11.30
sig ₂	230.8			231.7	225.9	350.5	231.1	223.0	430.5	233.2	219.0	538.7
sig ₃	9.79			9.81	9.84	2.80	9.80	8.13	8.61	9.81	9.12	8.94
sig ₄	282.0			282.0	275.5	283.1	282.0	269.4	277.8	282.0	253.5	253.9
MAE	0			0.421	3.637	28.07	0.505	4.628	51.84	3.935	6.575	83.64
LCor	1			0.999	0.999	0.992	0.999	0.999	0.971	0.999	0.999	0.915
RCor	1			1	1	0.904	1	1	0.738	1	1	0.833
	(5%, 0%)			(5%, 5%)			(5%, 10%)			(5%, 20%)		
$\hat{\Theta}_i$	SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC
b ₀	1044	1038	1004	1069	1037	1019	1069	1036	963	1078	1038	780
b ₁	93.30	94.33	93.79	87.57	90.04	52.82	88.07	90.90	45.29	86.52	90.75	80.88
b ₂	−46.69	−53.57	−49.05	−49.96	−42.94	−28.23	−52.68	−47.94	55.58	−32.18	−50.21	26.67
b ₃	1.04	4.58	6.55	8.27	1.27	8.63	9.70	6.75	−11.90	3.82	5.61	−15.79
sig ₁	15.19	13.65	16.59	15.25	17.02	8.34	15.36	13.93	16.19	15.79	12.80	4.10
sig ₂	231.0	223.7	213.6	229.8	218.7	332.4	228.9	216.0	412.7	232.4	211.9	518.8
sig ₃	9.94	7.93	11.07	9.97	8.63	5.05	9.95	7.52	10.61	10.11	8.42	6.56
sig ₄	282.4	281.4	347.6	282.6	275.0	349.4	282.6	268.7	342.2	282.6	253.2	317.6
MAE	1.146	2.646	16.78	4.970	5.516	34.43	5.089	6.287	62.03	8.118	8.108	88.97
LCor	0.999	0.999	0.996	0.999	0.999	0.991	0.999	0.999	0.971	0.999	0.999	0.911
RCor	1	1	1	1	1	0.928	1	1	0.738	1	1	0.809
	(10%, 0%)			(10%, 5%)			(10%, 10%)			(10%, 20%)		
$\hat{\Theta}_i$	SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC
b ₀	1070	1037	981	1069	1035	999	1070	1034	938	1080	1037	759
b ₁	86.59	94.05	90.52	87.58	90.27	50.902	88.10	90.80	46.43	86.29	90.33	81.31
b ₂	−45.96	−50.70	−62.01	−45.00	−39.50	−44.14	−47.79	−44.57	45.21	−28.34	−48.01	23.38
b ₃	6.79	5.26	12.25	6.69	1.46	14.45	8.13	6.78	−8.83	2.53	5.81	−16.26
sig ₁	14.80	24.43	4.36	14.97	27.35	25.00	15.15	22.09	30.51	15.93	21.39	18.64
sig ₂	228.7	220.1	205.1	230.0	215.3	323.1	229.1	212.5	401.1	232.5	208.7	506.5
sig ₃	9.79	11.46	5.94	9.89	12.10	6.336	9.88	10.16	13.87	10.11	10.50	8.73
sig ₄	282.4	279.6	398.0	282.5	273.2	389.6	282.5	267.2	379.8	282.5	251.8	354.3
MAE	5.608	4.699	30.12	5.387	8.279	40.15	5.424	8.191	68.85	8.747	9.636	93.20
LCor	0.999	0.999	0.990	0.999	0.999	0.987	0.999	0.999	0.969	0.999	0.999	0.908
RCor	1	1	0.904	1	1	0.976	1	1	0.833	1	1	0.833

Table 1. Cont.

$\hat{\Theta}_i$	(20%, 0%)			(20%, 5%)			(20%, 10%)			(20%, 20%)		
	SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC	SEM	SAEM	MCMC
b_0	1045	1035	896	1045	1044	908	1045	1036	857	1051	1039	687
b_1	93.34	94.05	100.8	93.70	80.78	64.43	94.05	90.14	57.35	92.78	89.41	91.59
b_2	−35.09	−38.36	2.54	−34.54	−128.87	21.66	−36.11	−34.79	97.19	−19.91	−39.24	85.55
b_3	−1.82	1.90	−16.60	−1.55	39.61	−13.36	−0.36	4.00	−31.09	−4.73	3.67	−42.98
sig_1	12.12	26.88	29.49	12.44	11.70	39.89	12.62	25.58	33.25	13.62	26.53	22.49
sig_2	230.9	211.4	204.7	231.8	118.2	315.9	231.3	204.4	387.9	233.1	200.9	485.3
sig_3	8.84	11.33	8.95	8.98	5.15	11.48	9.03	10.91	10.93	9.41	11.76	6.227
sig_4	282.3	276.2	465.9	282.3	265.3	453.6	282.3	264.0	442.2	282.4	249.2	411.6
MAE	3.261	8.002	57.07	3.374	32.83	67.38	2.994	10.905	93.01	6.343	12.47	117.2
LCor	0.999	0.999	0.969	0.999	0.991	0.969	0.999	0.999	0.944	0.999	0.998	0.873
RCor	1	1	0.976	1	0.9285	0.928	1	1	0.761	1	1	0.833

We marked in bold for each case the closest value to the reference among the three considered methods (the reference vector here was the simulated one that did not contain any missed value).

5.2. Real Data

In this section, we applied the three algorithms to the database used to create the simulated data. Of the 4500 RTs recorded during the experiment (30 participants producing the names of 150 items), 115 were removed by right censoring (2.56% of MNAR). Ref. [4] also removed 951 values, creating 21.13% of missing MCAR/MAR data. By applying the three proposed algorithms with some missing data to solve the problem in the presence of fixed and fixed–random interactions, we obtained the results presented in Table 2. The results were compared with respect to the initial vector where the missed values were treated by keeping the cases empty (KE). Interestingly, this is how missing and censored data are handled in psycholinguistic studies.

Table 2. Comparison of the initial parameters in the KE method versus the parameters estimated after the application of the SEM, SAEM and MCMC algorithms. The initial parameters presented here are the ones obtained from the model that contained 1066 NA (NA represent the incomplete data; here, we had 115 censored data and 951 missing data).

$\hat{\Theta}$	KE	SEM	SAEM	MCMC
b_0	852.5	875.7	892.334	1298
b_1	146.6	134.2	120.4	−161.4
b_2	158.6	126.1	79.63	−335.3
b_3	−58.12	−42.08	−26.72	146.6
sig_1	184.3	196.0	203.5	216.9
sig_2	133.0	127.8	94.87	221.05
sig_3	37.05	28.67	32.84	66.13
sig_4	279.5	302.5	262.9	516.6

6. Conclusions

In this study, we proposed an algorithm based on the stochastic expectation maximization (SEM) to estimate the parameters under a linear mixed-effects model (LMEM) that contained fixed–fixed and fixed–random interactions in the presence of different percentages of incomplete data.

The simulated and real data showed that the proposed SEM algorithm gave the best results compared to other competitors, where the bias of the parameters was smaller than the bias in the SAEM and MCMC algorithms.

The simulation results were obtained by using statistical software R (see Appendix A.5 for some source code).

We plan to extend this approach by considering a more complicated level of interaction such as random–random effects and the interactions between more than two variables. Another direction could be to consider an extension of the present work by considering a generalized model under the logistic regression by taking one for the NA values and zero for the observed ones.

Author Contributions: Conceptualization, C.P. and Y.S.; methodology, Y.S.; software, A.Z.; validation, A.Z., C.P. and Y.S.; formal analysis, A.Z.; investigation, C.P. and Y.S. resources, C.P.; data curation, C.P.; writing—original draft preparation, A.Z.; writing—review and editing, Y.S.; visualization, C.P. and Y.S.; supervision, C.P. and Y.S.; project administration, C.P. and Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data is unavailable due to privacy.

Acknowledgments: We would like to thank the Editor-in-Chief of Entropy, the associate Editor and two anonymous reviewers for their helpful comments, which helped us to focus on improving the original version of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In the following section, we present how [24] defined the models by including the maximum effects. It is interesting to show the types of missing data with the implementation methods. Then, the approach of Henderson is introduced and our model matrices to estimate the parameters are defined. Finally, we present some source code of the SEM algorithm.

Appendix A.1. Models Extension

Ref. [24] compared five types of models in order to prove the importance of taking the maximum effects. They presented the two random variables subject, designed by S , and items, designed by I . By taking i as the items' index, s as the subject's index and c as the observation, the first model was the maximum model that contained the fixed and random intercepts with the interactions, including also ρ_S and ρ_I :

$$Y_{c,i,s} = \beta_0 + S_{0,s} + I_{0,i} + (\beta_1 + S_{1,s} + I_{1,i})X_c + \epsilon_{c,i,s},$$

$$\begin{pmatrix} S_{0,s} \\ S_{1,s} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00}^2 & \rho_S \tau_{00} \tau_{11} \\ \rho_S \tau_{00} \tau_{11} & \tau_{11}^2 \end{pmatrix}\right),$$

$$\begin{pmatrix} I_{0,i} \\ I_{1,i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} w_{00}^2 & \rho_I w_{00} w_{11} \\ \rho_I w_{00} w_{11} & w_{11}^2 \end{pmatrix}\right),$$

$$\epsilon_{c,i,s} \sim (0, \sigma^2).$$

The second model differed by its correlation parameters, where $S_{0,s}$ and $S_{1,s}$ (respectively, $I_{0,i}$ and $I_{1,i}$) were independent. ρ_S (respectively, ρ_I) was set to zero:

$$Y_{c,i,s} = \beta_0 + S_{0,s} + I_{0,i} + (\beta_1 + S_{1,s} + I_{1,i})X_c + \epsilon_{c,i,s}, \quad (\text{A1})$$

$$\begin{pmatrix} S_{0,s} \\ S_{1,s} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00}^2 & 0 \\ 0 & \tau_{11}^2 \end{pmatrix}\right),$$

$$\begin{pmatrix} I_{0,i} \\ I_{1,i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} w_{00}^2 & 0 \\ 0 & w_{11}^2 \end{pmatrix}\right),$$

$$\epsilon_{c,i,s} \sim (0, \sigma^2).$$

In the third model, they reduced Equation (A1) by ignoring the items' random slopes (the interaction between the items' fixed and random effects), where w_{11} was set to 0:

$$Y_{c,i,s} = \beta_0 + S_{0,s} + I_{0,i} + (\beta_1 + S_{1,s})X_c + \epsilon_{c,i,s},$$

$$\begin{pmatrix} S_{0,s} \\ S_{1,s} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00}^2 & 0 \\ 0 & \tau_{11}^2 \end{pmatrix}\right),$$

$$I_{0,i} \sim (0, w_{00}^2),$$

$$\epsilon_{c,i,s} \sim (0, \sigma^2).$$

The fourth model kept the random items' slope and excluded that of the subject, where $\tau_{11} = 0$:

$$Y_{c,i,s} = \beta_0 + S_{0,s} + I_{0,i} + (\beta_1 + I_{1,i})X_c + \epsilon_{c,i,s},$$

$$S_{0,s} \sim (0, \tau_{00}^2),$$

$$\begin{pmatrix} I_{0,i} \\ I_{1,i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} w_{00}^2 & 0 \\ 0 & w_{11}^2 \end{pmatrix}\right),$$

$$\epsilon_{c,i,s} \sim (0, \sigma^2).$$

The fifth model excluded the two random slopes (for items and subject), where $\tau_{11} = w_{00} = 0$:

$$Y_{c,i,s} = \beta_0 + S_{0,s} + I_{0,i} + \beta_1 X_c + \epsilon_{c,i,s},$$

$$S_{0,s} \sim (0, \tau_{00}^2),$$

$$I_{0,i} \sim (0, w_{00}^2),$$

$$\epsilon_{c,i,s} \sim (0, \sigma^2).$$

In the next section, we present the different types of missing data and how to generate them using different methods.

Appendix A.2. Missing Data Types and Imputations

Let M be the missing data indicator that defines what is known and what is missing. The response variable is denoted by Y in the complete data, where $Y = \{Y_o, Y_{no}\}$ with Y_o is the observed part and Y_{no} is the nonobserved one. Ref. [48] denoted the missing part by Y_{miss} . Consider a dataset of p variables and n subjects, with $Y = \{Y_1, \dots, Y_p\}$. Y should look like a matrix of n rows and p columns. In the literature on missing data, see also [49,50].

We show in this section the three types of missing values and how to generate them in a simulation study based on a rule in each case ([51]).

We start with the first type named MCAR.

The missing completely at random type (MCAR) is a special case of the MAR type that is presented later. In this type, the value is missed due to chance and the absence is not related to the subject, so the distribution of M does not depend on Y_o or on Y_{miss} and is identical for all the observations:

$$\mathbb{P}(M|Y) = \mathbb{P}(M),$$

The statistical advantage of MCAR data is that the analysis remains unbiased, none of the variables is affected more than another.

Suppose that each subject has a probability π of being missed from a variable, the missing data rule is given by $\mathbb{P}(M = 1) = \pi$. From this rule, we can determine various properties associated with the MCAR data such as the expected percentage of missing values and the expected number of missing data patterns in the MCAR data.

If we are working with univariate patterns, let n be the number of subjects in the data and K the random variable indicating the number of subjects with missing values in the data that follows a binomial distribution given by $K \sim B(n, \pi)$, where $0 \leq \pi \leq 1$. Since $\mathbb{E}(K) = n\pi$ and $\mathbb{V}(K) = n\pi(1 - \pi)$, the expected percentage of missing values is:

$$\begin{aligned}\mathbb{E}(\Pi) &= \frac{1}{n}\mathbb{E}(K) \\ &= \pi,\end{aligned}$$

where $\Pi = \frac{K}{n}$ is the random variable that defines the estimated percentage of missing values in a sample. the variance of this estimated percentage is given by:

$$\begin{aligned}\mathbb{V}(\Pi) &= \frac{1}{n^2}\mathbb{V}(K) \\ &= \frac{\pi(1 - \pi)}{n}.\end{aligned}$$

In this case, we have two missing data patterns, pattern 1 includes subjects with complete data; pattern 2 includes subjects with missing data. Let I_j be the indicator variable of the event, where pattern j is present in at least one subject in the sample with $j \in \{1, 2\}$. The probability that pattern 1 is present in at least one subject is $\mathbb{P}(I_1 = 1) = \mathbb{E}(I_1) = 1 - \pi^n$. The probability that pattern 2 is present in at least one subject is $\mathbb{P}(I_2 = 1) = \mathbb{E}(I_2) = 1 - (1 - \pi)^n$. Let D be the number of distinct missing data patterns, $D = \sum_{j=1}^2 I_j$; we can determine the expected value:

$$\begin{aligned}\mathbb{E}(D) &= \sum_{j=1}^2 \mathbb{E}(I_j) \\ &= (1 - \pi^n) + (1 - (1 - \pi)^n) \\ &= 2 - \pi^n - (1 - \pi)^n.\end{aligned}\tag{A2}$$

Therefore, in MCAR data that contain more than two missing data patterns, the expected number of distinct pattern in a sample is given by:

$$\mathbb{E}(D) = m - \sum_{j=1}^m (1 - \eta_j)^n,$$

with η_1, \dots, η_m are the corresponding probabilities for patterns $1, \dots, m$, and depends only on the probability of missing values in each variable i (i.e., π_1, \dots, π_l). $m = 2^l$ is the number of patterns and l is the total number of variables. Generally, the most used method to implement the MCAR type is randomly deleting the desired percentage of missing values.

The second type is MAR.

Missing at random is the most classic case; Rubin and D.B (1976) defined missing data to be MAR if the distribution of missing data did not depend on Y_{miss} :

$$\mathbb{P}(M|Y) = \mathbb{P}(M|Y_o).$$

MAR data occur when the absence is not random but can be explained by the variables for which complete information exists. More generally, we are in the MAR type if the cause of missing values is not related to the goal of the study. If Y_1 has missing values, then it is regressed on other variables Y_2 to Y_l . The missing values in Y_1 are then replaced by the obtained predictive values. Similarly, if Y_2 has missing values, then the Y_1, Y_3 to Y_p variables are used in the prediction model as independent variables. Thus, the missing values are replaced with the predicted ones. The missing data rules for MAR data can be organized into several categories: single-cutoff method, multiple-cutoff method, percentile method and logistic regression method.

Here, we define these three rules.

Single-cutoff method: Consider Y_1 as the variable with missing value and Y_2 is the missing data predictor with a cutoff point a . If a subject has $Y_2 \geq a$, then its probability of being missing from Y_1 is π_1 , and if $Y_2 < a$, then its probability of being missing from Y_1 is π_2 . Let U be an indicator that takes the value 1 ($U = 1$) when $Y_2 \geq a$ and $U = 0$ when $Y < a$. The missing data rule is given as:

$$\mathbb{P}(M = 1|U = 1) = \pi_1; \mathbb{P}(M = 1|U = 0) = \pi_2,$$

where the parameters associated with this rule are π_1 and π_2 , and π_0 is the probability that Y_2 is equal to or greater than the cutoff: $\mathbb{P}(Y_2 \geq a) = \mathbb{P}(U = 1) = \pi_0$. To find the expected percentage of missing values, we first calculate the unconditional probability of a subject being missing from Y_1 :

$$\begin{aligned}\pi_{miss} &= \mathbb{P}(M = 1) \\ &= \mathbb{P}(M = 1|U = 1)\mathbb{P}(U = 1) + \mathbb{P}(M = 1|U = 0)\mathbb{P}(U = 0) \\ &= \pi_1\pi_0 + \pi_2(1 - \pi_0).\end{aligned}$$

The expected percentage of missing values in a sample is given by:

$$\begin{aligned}\mathbb{E}(\Pi) &= \pi_{miss} \\ &= \pi_1\pi_0 + \pi_2(1 - \pi_0),\end{aligned}$$

and the variance of this estimated percentage of missing values is written as:

$$\mathbb{V}(\Pi) = \frac{\pi_{miss}(1 - \pi_{miss})}{n}.$$

Multiple-cutoff method: One advantage of this method is that it can be used to create a nonlinear relationship between the missing data indicator and the missing data predictor. To create a nonlinear relationship between the indicator and the predictor, we need to fix an upper cutoff and a lower cutoff. Suppose we fix the two cutoff points a and $-a$, $U = 1$ when $Y_2 \geq a$ or $Y_2 \leq -a$, and $U = 0$ when $-a < Y_2 < a$. The missing data rule can be written as:

$$\mathbb{P}(M = 1|U = 1) = \pi_1; \mathbb{P}(M = 1|U = 0) = \pi_2, \quad (\text{A3})$$

and we can see that this missing data rule is the same as the missing data rule in the single-cutoff method. In the case of a linear relationship, we need to specify two or more cutoff points in the missing data predictor. Suppose we take the quartile points (Q_1 , Q_2 and Q_3) as a cutoff, let V be a discrete uniform random variable created based on the values of Y_2 and that takes the following values:

$$V = \begin{cases} 1 & \text{if } Y_2 < Q_1 \\ 2 & \text{if } Q_1 \leq Y_2 < Q_2 \\ 3 & \text{if } Q_2 \leq Y_2 < Q_3 \\ 4 & \text{if } Y_2 \geq Q_3. \end{cases} \quad (\text{A4})$$

Then, for the linear relation case, the missing data rule is given by:

$$\begin{aligned}\mathbb{P}(M = 1|V = 1) &= \pi_1; \mathbb{P}(M = 1|V = 2) = \pi_2; \\ \mathbb{P}(M = 1|V = 3) &= \pi_3; \mathbb{P}(M = 1|V = 4) = \pi_4.\end{aligned}$$

Let n be the total number of subjects and n_j be the number of subjects in each quartile group. The variance of the estimated π_0 is:

$$\mathbb{V}(\Pi_0) = \frac{\pi_0(1 - \pi_0)}{n},$$

and the variance of the estimated π_j where $j \in \{1, 2, 3, 4\}$ is defined as follows:

$$\mathbb{V}(\Pi_j) = \frac{\pi_j(1 - \pi_j)}{n_j}.$$

We can calculate each subject's probability of being missing by calculating the marginal probability of $M = 1$:

$$\begin{aligned}\pi_{miss} &= \mathbb{P}(M = 1) \\ &= \pi_1\pi_0 + \pi_2\pi_0 + \pi_3\pi_0 + \pi_4\pi_0,\end{aligned}$$

and the expected percentage of missing values can be written as:

$$\begin{aligned}\mathbb{E}(\Pi) &= \pi_{miss} \\ &= \pi_1\pi_0 + \pi_2\pi_0 + \pi_3\pi_0 + \pi_4\pi_0;\end{aligned}$$

thus, the variance of this estimated percentage is given by:

$$\mathbb{V}(\Pi) = \frac{\pi_{miss}(1 - \pi_{miss})}{n}.$$

To implement the missing data rule in (A4), researchers typically delete π_1 subjects with $Y_2 < Q_1$, π_2 subjects with $Q_1 \leq Y_2 < Q_2$, and so on.

Percentile method: This case is an extension of the multiple-cutoff method, where each subject has a probability of being missing that depends on its percentile rank in the missing data predictor.

If there is a direct relation between the missing data indicator and the predictor, then the missing data rule is defined by:

$$\mathbb{P}(M = 1|Y_2 = q_k) = \frac{k}{100},$$

where q_k is the Y_2 value corresponding to its k th percentile. If there is an indirect relationship, then the missing data rule is defined as:

$$\mathbb{P}(M = 1|Y_2 = q_k) = 1 - \frac{k}{100}.$$

Logistic regression method: When generating MAR data by using the logistic regression method, the logistic regression model is considered as the missing data rule and the population regression coefficients of the model are the parameters associated with the missing data rule. Then, the logistic regression model for subject i is written as:

$$\log \left(\frac{\mathbb{P}(M_i = 1|y_{2,i})}{1 - \mathbb{P}(M_i = 1|y_{2,i})} \right) = \beta_0 + \beta_1 y_{2,i},$$

where $y_{2,i}$ is the subject i 's value on Y_2 . The probability of being missing for each subject is given by:

$$\mathbb{P}(M_i = 1|y_{2,i}) = \frac{1}{1 + e^{-\beta_0 - \beta_1 y_{2,i}}}.$$

Because the above function is continuous, it means the probability of being missing for Y_1 gradually increases or decreases as the value of Y_2 increases. With the logistic regression,

there is no simple formula for calculating the expected percentage of missing data, so we estimate the expected percentage of missing data by calculating the mean of the probabilities in a sample with a large sample size:

$$\pi_{miss} = \frac{1}{n} \sum_{i=1}^n \frac{1}{1 + e^{-\beta_0 - \beta_1 y_{2,i}}}.$$

The third and last type is the MNAR data:

In missing not at random data, the cause of missing data is related to the variable of interest. In this type, the probability that an observation is missing depends on the observed Y_o and the missing data Y_{miss} . If Y_1 is neither MCAR nor MAR, then it is MNAR. Generating MNAR data is the same as generating MAR data; the only difference is that in MNAR data the probability of missing values depends on the variable's own value and not on the observed values of the other variables. Therefore, we can change the missing data predictor to the variable with missing values, then we use one of the methods presented above to generate MAR data. For example, to change rule (A3) that generates MAR data to one that generates MNAR data, we only need to replace the variable Y_2 by Y_1 ; therefore, the corresponding missing data rule for generating MNAR data is when Y_1 value is above a cutoff point a . Y_1 has a π_1 probability of being missing, otherwise, Y_1 has a π_2 probability of being missing.

Next, we define the linear system of Henderson to deal with the parameters' estimation.

Appendix A.3. Henderson's Approach

In order to estimate the parameters $\hat{\delta}$ and \hat{u} , we need to maximize the log function $(\hat{\delta}, \hat{u}) = \arg \max_{\delta, u} \ln f(Y, u)$. We define $f(Y, u) = f(Y|u)f(u)$ with $Y|u \sim \mathcal{N}(X\delta + Zu, R)$ and $u \sim \mathcal{N}(0, G)$, so we can write:

$$\begin{aligned} -2 \ln f(Y|u) &= N \ln(2\pi) + \ln |R| + (Y - X\delta - Zu)' R^{-1} (Y - X\delta - Zu), \\ -2 \ln f(u) &= q \ln(2\pi) + \ln |G| + u' G^{-1} u. \end{aligned}$$

Denote $l(\delta, u; Y) = \ln f(Y, u)$; to maximize this function, we need to cancel the first derivatives of $\ln f(Y, u)$ with respect to δ and u :

$$\begin{aligned} \frac{\partial [-2l(\delta, u; y)]}{\partial \delta} &= -2X' R^{-1} (y - X\delta - Zu) = 0, \\ \frac{\partial [-2l(\delta, u; y)]}{\partial u} &= -2Z' R^{-1} (y - X\delta - Zu) + 2G^{-1} u = 0. \end{aligned} \quad (A5)$$

From the two equations in (A5), Henderson presented the linear mixed model under the system:

$$\begin{bmatrix} X' R^{-1} X & X' R^{-1} Z \\ Z' R^{-1} X & Z' R^{-1} Z + G^{-1} \end{bmatrix} \begin{bmatrix} \hat{\delta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X' R^{-1} y \\ Z' R^{-1} y \end{bmatrix} \quad (A6)$$

In the next part, we present how we applied Henderson's approach to our proposed model.

Appendix A.4. Model Simplification

Complete model: We consider the following matrices, where r determines the rows, c determines the columns, and $l, h \in [1, p+k]$:

$$(X' R^{-1} X)_{M \times N} = \begin{cases} (n-p-k) + \sigma_{e_1}^{-2} + \dots + \sigma_{e_p}^{-2} + \sigma_{e_{p+1}}^{-2} + \dots + \sigma_{e_{p+k}}^{-2} & ; \text{if } r = c = 1 \\ \sum_{j=1}^{n-p-k} x_{l,j} + \sum_{o=1}^{p+k} x_{l,n-p-k+o} \sigma_{e_o}^{-2} & ; \text{if } r = 1; c \neq 1 \\ \sum_{j=1}^{n-p-k} x_{l,j} x_{h,j} + \sum_{o=1}^{p+k} x_{l,n-p-k+o} x_{h,n-p-k+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(X'R^{-1}y)_{M \times 1} = \begin{cases} \sum_{j=1}^{n-p-k} y_j + \sum_{o=1}^{p+k} y_{n-p-k+o} \sigma_{e_o}^{-2} & ; \text{if } r = 1 \\ \sum_{j=1}^{n-p-k} x_{l,j} y_j + \sum_{o=1}^{p+k} x_{l,n-p-k+o} y_{n-p-k+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(X'R^{-1}Z)_{M \times N} = \begin{cases} \sum_{j=1}^{n-p-k} z_{l,j} + \sum_{o=1}^{p+k} z_{l,n-p-k+o} \sigma_{e_o}^{-2} & ; \text{if } r = 1 \\ \sum_{j=1}^{n-p-k} x_{l,j} z_{h,j} + \sum_{o=1}^{p+k} x_{l,n-p-k+o} z_{h,n-p-k+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(Z'R^{-1}Z)_{M \times N} = \begin{cases} \sum_{j=1}^{n-p-k} z_{l,j}^2 + \sum_{o=1}^{p+k} z_{l,n-p-k+o}^2 \sigma_{e_o}^{-2} & ; \text{if } r = c \\ \sum_{j=1}^{n-p-k} z_{l,j} z_{h,j} + \sum_{o=1}^{p+k} z_{l,n-p-k+o} z_{h,n-p-k+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(Z'R^{-1}y)_{M \times 1} = \left[\sum_{j=1}^{n-p-k} z_{l,j} y_j + \sum_{o=1}^{p+k} z_{l,n-p-k+o} y_{n-p-k+o} \sigma_{e_o}^{-2} \right].$$

Case of no interactions: In this case, all the matrices of the linear system are simplified to reach p instead of $p+k$. $\forall l, h \in [1, p]$:

$$(X'R^{-1}X)_{M \times N} = \begin{cases} (n-p) + \sigma_{e_1}^{-2} + \dots + \sigma_{e_p}^{-2} & ; \text{if } r = c = 1 \\ \sum_{j=1}^{n-p} x_{l,j} + \sum_{o=1}^p x_{l,n-p+o} \sigma_{e_o}^{-2} & ; \text{if } r = 1, c \neq 1 \\ \sum_{j=1}^{n-p} x_{l,j} x_{h,j} + \sum_{o=1}^p x_{l,n-p+o} x_{h,n-p+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(X'R^{-1}y)_{M \times 1} = \begin{cases} \sum_{j=1}^{n-p} y_j + \sum_{o=1}^p y_{n-p+o} \sigma_{e_o}^{-2} & ; \text{if } r = 1 \\ \sum_{j=1}^{n-p} x_{l,j} y_j + \sum_{o=1}^p x_{l,n-p+o} y_{n-p+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(X'R^{-1}Z)_{M \times N} = \begin{cases} \sum_{j=1}^{n-p} z_{l,j} + \sum_{o=1}^p z_{l,n-p+o} \sigma_{e_o}^{-2} & ; \text{if } r = 1 \\ \sum_{j=1}^{n-p} x_{l,j} z_{h,j} + \sum_{o=1}^p x_{l,n-p+o} z_{h,n-p+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(Z'R^{-1}Z)_{M \times N} = \begin{cases} \sum_{j=1}^{n-p} z_{l,j}^2 + \sum_{o=1}^p z_{l,n-p+o}^2 \sigma_{e_o}^{-2} & ; \text{if } r = c \\ \sum_{j=1}^{n-p} z_{l,j} z_{h,j} + \sum_{o=1}^p z_{l,n-p+o} z_{h,n-p+o} \sigma_{e_o}^{-2} & ; \text{otherwise.} \end{cases}$$

$$(Z'R^{-1}y)_{M \times 1} = \left[\sum_{j=1}^{n-p} z_{l,j} y_j + \sum_{o=1}^p z_{l,n-p+o} y_{n-p+o} \sigma_{e_o}^{-2} \right].$$

Appendix A.5. R Source Code

Finally, we give the source code in R for the SEM algorithm by considering our real data.

```
library(lme4); # for lmer
library(msm); # for rtnorm
# initial model
model_initial=lmer(RT~AoA*Lett_cat+(1|AoA|participants)+(1|items), data,
control = lmerControl(optimizer="bobyqa", calc.derivs = FALSE));

# random initialization of parameters
b0=fixef(model_initial)[1];
b1=fixef(model_initial)[2];
b2=fixef(model_initial)[3];
b3=fixef(model_initial)[4];
```

```

sig_items=attr(VarCorr(model_initial)$items,"stddev");
sig_parts=attr(VarCorr(model_initial)$participants,"stddev")[1];
sig_parts_conver<-rep(0.0,itermax);
sig_parts.AoA=attr(VarCorr(model_initial)$participants,"stddev")[2];
sig_Err=attr(VarCorr(model_initial),"sc");

Err=rnorm(n, mean=0, sd=sig_Err);
Rand_parts=rep(0,n);
Rand_items=rep(0,n);
Rand_parts.AoA=rep(0,n);
Rand_parts=rep(rnorm(np,mean=0, sd=sig_parts),each=ni);
Rand_items=rep(rnorm(ni,mean=0, sd=sig_items),each=np);
Rand_parts.AoA=rep(rnorm(np,mean=0, sd=sig_parts.AoA),each=ni);

Lett_cat_1=vector(length=n)
for(i in 1:n){
  Lett_cat_1[i]<-ifelse(test=(Lett_cat[i]==Lett_cat[2]),1,0)
}

u=runif(1,min=0,max=1);
# acceptance prob based on the Hamiltonian MC
p<-function(x,newx){
  a=min(1,exp(x-newx));
  if(u<a){
    x=newx;
  }
  else{
    x=x;
  }
}

# SEM (Algorithm 1)
itermax=50; # max number of iteration for SEM
burnin=10; # burnin length
GS=5; # number of iterations for Gibbs~sampling

# main loop
for(iter in 1:itermax){
  # fixed part
  fixed=b0+b1*AoA+b2*Lett_cat_1+b3*AoA*Lett_cat_1;
  # Gibbs sampling
  for(j in 1:GS) {
    # Err with constraints
    newErr=rtnorm(length(Err),sd=sig_Err,
    lower=min(Err)-fixed-Rand_parts-Rand_items-Rand_parts.AoA,
    upper=max(Err)-fixed-Rand_parts-Rand_items-Rand_parts.AoA);
    p(Err,newErr);
    # Rand_parts with constraints
    for (a in 1:length(factor(participants))){
      Subset1=(participants == participants[a]) & censored;
      newRand_parts=Rand_parts;
      if (sum(Subset1)>0) {
        low=max(min(!is.na(data[participants == participants[a],5]))-Rand_items[Subset1]
        -Rand_parts.AoA[Subset1]-Err[Subset1]-fixed[Subset1]);

```

```

    upp=min(max(!is.na(data[participants == participants[a],5]))-Rand_items[Subset1]
    -Rand_parts.AoA[Subset1]-Err[Subset1]-fixed[Subset1]);
    upp=ifelse(upp-low>0,upp,upp+abs(upp-low)+100);
    newRand_parts[Subset1]=rtnorm(1,sd=sig_parts, lower=low,upper=upp);
  }
}
p(Rand_parts,newRand_parts);
# Rand_items with constraints
for (a in 1:length(factor(items))){
  Subset2=(items == items[a]) & censored;
  newRand_items=Rand_items;
  if (sum(Subset2)>0){
    low=max(min(!is.na(data[items == items[a],5]))-Rand_parts[Subset2]
    -Rand_parts.AoA[Subset2]-Err[Subset2]-fixed[Subset2]);
    upp=min(max(!is.na(data[items == items[a],5]))-Rand_parts[Subset2]
    -Rand_parts.AoA[Subset2]-Err[Subset2]-fixed[Subset2]);
    upp=ifelse(upp-low>0,upp,upp+abs(upp-low)+100);
    newRand_items[Subset2]=rtnorm(1,sd=sig_items, lower=low,upper=upp);
  }
}
p(Rand_items,newRand_items);
# Rand_parts.AoA with constraints
for (a in 1:length(factor(participants))){
  Subset1=(participants == participants[a]) & censored;
  newRand_parts.AoA=Rand_parts.AoA;
  if (sum(Subset1)>0) {
    low=max(min(!is.na(data[participants == participants[a],5]))-Rand_items[Subset1]
    -Rand_parts[Subset1]-Err[Subset1]-fixed[Subset1]);
    upp=min(max(!is.na(data[participants == participants[a],5]))-Rand_items[Subset1]
    -Rand_parts[Subset1]-Err[Subset1]-fixed[Subset1]);
    upp=ifelse(upp-low>0,upp,upp+abs(upp-low)+100);
    newRand_parts.AoA[Subset1]=rtnorm(1,sd=sig_parts.AoA, lower=low,upper=upp);
  }
}
p(Rand_parts.AoA,newRand_parts.AoA);
# end of Gibbs sampling algorithm
}
# compute the new response vector
RT[censored]=round(fixed[censored]+Rand_parts[censored]+Rand_items[censored]
+Rand_parts.AoA[censored]+Err[censored],0);
# train the model with the complete data
model_final<-lmer(RT~AoA*Lett_cat+(1|AoA|participants)+(1|items), data,
control = lmerControl(optimizer="bobyqa",calc.derivs = FALSE));
# end of SEM algorithm
}

```

References

1. Donders, F.C. On the speed of mental processes. *Acta Psychol.* **1969**, *30*, 412–431. [[CrossRef](#)] [[PubMed](#)]
2. Roelofs, A. One hundred fifty years after Donders: Insights from unpublished data, a replication, and modeling of his reaction times. *Acta Psychol.* **2018**, *191*, 228–233. [[CrossRef](#)] [[PubMed](#)]
3. Baayen, R.H.; Davidson, D.J.; Bates, D.M. Mixed-effects modeling with crossed random effects for subjects and items. *J. Mem. Lang.* **2008**, *29*, 390–412. [[CrossRef](#)]
4. Perret, C.; Solier, C. Application of a Bayesian approach for exploring the impact of syllable frequency in handwritten picture naming. *J. Cogn. Psychol.* **2022**, *34*, 622–635. [[CrossRef](#)]

5. Ratcliff, R. Methods for dealing with reaction time outliers. *Psychol. Bull.* **1993**, *114*, 510–532. [\[CrossRef\]](#)
6. Van Selst, M.; Jolicoeur, P. A solution to the effect of the size on outlier elimination. *Q. J. Exp. Psychol.* **1994**, *47*, 631–650. [\[CrossRef\]](#)
7. Barr, D.J.; Levy, R.; Scheepers, C.; Tily, H.J. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *J. Mem. Lang.* **2013**, *68*, 255–278. [\[CrossRef\]](#)
8. Dempster, A.P.; Rubin, D.B.; Tsutakawa, R.K. Estimation in Covariance Components Models. *J. Amer. Statist. Assoc.* **1981**, *76*, 341–353. [\[CrossRef\]](#)
9. West, T.B.; Welch, B.K.; Galecki, T.A. *Linear Mixed Models, A Practical Guide Using Statistical Software*; Chapman & Hall: New York, NY, USA, 2006.
10. Jiang, J.; Wand, M.P.; Bhaskaran, A. Usable and precise asymptotics for generalized linear mixed model analysis and design. *J. R. Stat. Soc. Ser. B* **2022**, *84*, 55–82. [\[CrossRef\]](#)
11. Faraway, J.J. *Extending the Linear Model with R*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2016.
12. Jiang, J. *Asymptotic Analysis of Mixed Effects Models*; CRC Press: Boca Raton, FL, USA, 2017.
13. McCulloch, C.E.; Searle, S.R.; Neuhaus, J.M. *Generalized, Linear, and Mixed Models*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2008.
14. Stroup, W.W. *Generalized Linear Mixed Models*; CRC Press: Boca Raton, FL, USA, 2013.
15. Bates, D.; Mächler, M.; Bolker, M.B.; Walker, C.S. Fitting Linear Mixed-Effects Models Using lme4. *J. Stat. Softw.* **2015**, *67*, 1–48. [\[CrossRef\]](#)
16. Boik, J.R. The Analysis of Two-Factor Interactions in Fixed Effects Linear Models. *J. Educ. Stat.* **1993**, *18*, 1–40. [\[CrossRef\]](#)
17. Yi, Q.; Panzarella, T. *Estimating Sample Size for Tests on Trends across Repeated Measurements with Missing Data Based on the Interaction Term in a Mixed Model*; Elsevier: Amsterdam, The Netherlands, 2002; Volume 23, pp. 481–96.
18. Robin, G.; Josse, J.; Moulines, E.; Tibshirani, R. Main effects and interactions in mixed and incomplete data frames. *J. Am. Statist. Assoc.* **2019**, *115*, 1292–1303. [\[CrossRef\]](#)
19. Giesselmann, M.; Schmidt-Catran, A.W. Interactions in fixed effects regression models. *Sociol. Methods. Res.* **2022**, *51*, 1100–1127. [\[CrossRef\]](#)
20. Martin, D.A.; Quinn, M.K.; Park, H.J. MCMCpack: Markov chain Monte Carlo in R. *J. Stat. Softw.* **2011**, *42*, 22. [\[CrossRef\]](#)
21. Hadfield, J.D. MCMC methods for multi-response generalized linear mixed models: The MCMCglmm R package. *J. Stat. Softw.* **2010**, *33*, 1–22. [\[CrossRef\]](#)
22. Khodadadian, A.; Parvizi, M.; Teshnehlal, M.; Heitzinger, C. Rational Design of Field-Effect Sensors Using Partial Differential Equations, Bayesian Inversion, and Artificial Neural Networks. *Sensors* **2022**, *22*, 4785. [\[CrossRef\]](#)
23. Noii, N.; Khodadadian, A.; Ulloa, J.; Aldakheel, F.; Wick, T.; Francois, S.; Wriggers, P. Bayesian Inversion with Open-Source Codes for Various One-Dimensional Model Problems in Computational Mechanics. *Arch. Comput. Methods Eng.* **2022**, *29*, 4285–4318. [\[CrossRef\]](#)
24. Matuschek, H.; Kliegl, R.; Vasishth, S.; Baayen, H.; Bates, D. Balancing Type I error and power in linear mixed models. *J. Mem. Lang.* **2017**, *94*, 305–315. [\[CrossRef\]](#)
25. Dempster, A.; Laird, N.; Rubin, D. Maximum-likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. Ser. B* **1977**, *39*, 1–38.
26. Delyon, B.; Lavielle, M.; Moulines, E. Convergence of a stochastic approximation version of the EM algorithm. *Ann. Stat.* **1999**, *1*, 94–128. [\[CrossRef\]](#)
27. Lavielle, M.; Mbogning, C. An improved SAEM algorithm for maximum likelihood estimation in mixtures of non linear mixed effects models. *Statist. Comput.* **2014**, *24*, 693–707. [\[CrossRef\]](#)
28. Robbins, H.; Monro, S. A Stochastic Approximation Method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [\[CrossRef\]](#)
29. Bhatnagar, S.; Prasad, H.; Prashanth, L. *Stochastic Recursive Algorithms for Optimization*; Springer: Heidelberg, Germany, 2013; Volume 434, pp. 17–28.
30. Celeux, G.; Diebolt, J. The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Comput. Stat. Q.* **1985**, *2*, 73–82.
31. Wei, G.; Tanner, M. A Monte-Carlo implementation of the EM algorithm and the Poor's Man's data augmentation algorithm. *J. Am. Statist. Assoc.* **1990**, *85*, 699–704. [\[CrossRef\]](#)
32. Bennett, D.A. How can I deal with missing data in my study. *Aust. N. Z. J. Public Health* **2001**, *25*, 464–469. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Slaoui, Y.; Nuel, G. Parameter Estimation in a Hierarchical Random Intercept Model with Censored Response: An Approach using a SEM Algorithm and Gibbs Sampling. *Sankhya Indian J. Statist.* **2014**, *76*, 210–233. [\[CrossRef\]](#)
34. Foulley, J.L. Mixed Model Methodology. Part I: Linear Mixed Models. In *Technical Report*; Université de Montpellier: Montpellier, France, 2015.
35. Grund, S.; Lüdtke, O.; Robitzsch, A. Multiple imputation of missing data in multilevel models with the R package mdmb: A flexible sequential modeling approach. *Behav. Res. Meth.* **2021**, *53*, 2631–2649. [\[CrossRef\]](#)
36. Santos, F. *L'algorithme EM: Une Courte Présentation*; CNRS, UMR, 5199: Talence, France, 2015.
37. Celeux, G.; Diebolt, J. Une Version de Type Recuit Simule de L'algorithme EM. Ph.D. Thesis, INRIA, Le Chesnay, France, 1989.
38. Celeux, G.; Chauveau, D.; Diebolt, J. On Stochastic Versions of the EM Algorithm. Ph.D. Thesis, INRIA, Le Chesnay, France, 1995.
39. Celeux, G.; Diebolt, J. A stochastic approximation type EM algorithm for the mixture problem. *Stochastics Int. J. Probab. Stoch. Process.* **1992**, *41*, 119–134. [\[CrossRef\]](#)

40. Richard, F.; Samson, A.; Cuénod, A.C. A SAEM algorithm for the estimation of template and deformation parameters in medical image sequences. *Statist. Comput.* **2009**, *19*, 465–478. [[CrossRef](#)]
41. Panhart, X.; Samson, A. Extension of the SAEM algorithm for nonlinear mixed models with 2 levels of random effects. *Biostatistics* **2008**, *10*, 121–135. [[CrossRef](#)]
42. Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv* **2017**, arXiv:1701.02434.
43. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [[CrossRef](#)]
44. Hastings, W.K. Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika* **1970**, *57*, 97–109. [[CrossRef](#)]
45. Roberts, O. G.; Rosenthal, S. J. General state space Markov chains and MCMC algorithms. *Probab. Surv.* **2004**, *1*, 20–71. [[CrossRef](#)]
46. Andrieu, C.; De Freitas, N.; Doucet, A.; Jordan, M. An Introduction to MCMC for Machine Learning. *KAP* **2003**, *50*, 5–43.
47. Perret, C.; Bonin, P. Which variables should be controlled for to investigate picture naming in adults? A Bayesian meta-analysis. *Behav. Res. Methods* **2019**, *51*, 2533–2545. [[CrossRef](#)]
48. Schafer, L.J.; Graham, W.J. Missing Data: Our View of the State of the Art. *Psychol. Methods* **2002**, *7*, 147–177. [[CrossRef](#)]
49. Enders, C.K. *Applied Missing Data Analysis*; Guilford Press: New York, NY, USA, 2010.
50. Little, R.J.; Rubin, D.B. *Statistical Analysis with Missing Data*; John Wiley & Sons: Hoboken, NJ, USA, 2019; Volume 793.
51. Zhang, X. Tutorial: How to Generate Missing Data For Simulation Studies. 2021. Available online: <https://files.osf.io/v1/resources/rq6yb/providers/osfstorage/60e93adf600da501810a8ea4?format=pdf&action=download&direct&version=1> (accessed on 8 February 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.