

# Efficient Maximum Likelihood estimation in large-scale multilevel models

December 6, 2006

Prepared by:

Murray Aitkin and Irit Aitkin  
School of Behavioural Science  
University of Melbourne

Prepared for:

US Department of Education  
Office of Educational Research and Improvement  
National Center for Education Statistics

This project was an activity of the NAEP Education Statistics Services Institute.

# Contents

<b>1</b>	<b>Aim of the project</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
<b>3</b>	<b>The assessment of current packages</b>	<b>5</b>
3.1	Gllamm . . . . .	5
3.2	xtmixed in Stata . . . . .	5
3.3	Latent Gold . . . . .	6
3.4	VARCL . . . . .	7
3.5	MLWin . . . . .	7
3.6	HLM . . . . .	7
3.7	R . . . . .	7
3.8	SAS Proc NLMIXED . . . . .	8
3.9	WinBUGS . . . . .	8
<b>4</b>	<b>Extensions to the EM algorithm</b>	<b>8</b>
4.1	Extension of the EM algorithm for two-level GLMMs to four levels . . .	8
4.2	The redesign of the M step of the algorithm to reduce the effective data size for this step . . . . .	9
4.3	The parallelizing of the numerical integration step at each random-effect level . . . . .	10
4.4	The acceleration of the M step by expanding the step-length . . . . .	11
4.5	The inclusion of incomplete covariate data by computing a close approximation to the full information matrix for the incomplete data . . . . .	11
<b>5</b>	<b>Bayesian methods</b>	<b>12</b>
<b>6</b>	<b>Summary and developments proposed</b>	<b>14</b>

# 1 Aim of the project

The aim of this project was to investigate efficient methods for maximum likelihood (ML) estimation in large-scale multilevel models, with particular reference to NAEP-scale national and international educational achievement surveys with binary and multi-category response items. The investigation had three strands:

- the assessment of current packages for logit latent variable models, for their possible extension to NAEP-scale data structures;
- a detailed investigation of extensions to the EM algorithm for fitting these models to NAEP-scale data structures;
- an investigation of other methods for these models, including Bayesian MCMC methods.

## 2 Background

Statistical modelling and analysis is being increasingly applied to systems with multiple model levels, ranging from education (e.g., classes, schools, districts) to clinical data on patient outcomes (e.g., clinics, hospitals, regions and states). While the underlying theory behind multilevel models was developed 25 years ago, efficient algorithmic and programming implementation of the models took much longer, and has lagged behind both theory and applications.

The multilevel structures of interest in education surveys are generally *hierarchical* or *nested*: in national testing in schools, students are nested in schools (or in classes within schools) and schools are nested in higher-level administrative areas, which may be nested in states. In international surveys there is a further level of country nesting. In health surveys, for example of adverse events following specific surgery types, events may be nested within practitioners or clinics, nested in hospitals, nested in state or other administrative areas.

Not all structures of interest are pure hierarchies. In school studies, students in high school may come from catchment areas outside the local neighborhood of the high school, so there may be “neighborhood effects” which are *crossed*, or partly crossed and partly nested, with high school effects, leading to more complex models. However the survey designs used in the NAEP surveys are fully nested (students within schools within PSUs), so the hierarchical model is directly appropriate for these survey structures.

The analysis of nested or hierarchical data sets aims to relate outcomes at the “lowest” level – student item scores – to explanatory variables at every level of the hierarchy, since student item scores are potentially affected by student ability, family environment, teacher qualities, school equipment and climate, and state curriculum design.

The statistical models for hierarchical structures use *random effect* or *variance component* models to represent the random variation at each level of the structure. Maximum likelihood analysis of these models relies heavily on the *EM algorithm* (Dempster, Laird and Rubin 1977), though other approaches, particularly Gauss-Newton and Markov Chain Monte Carlo (MCMC) simulation methods, are possible and have become popular.

The EM algorithm for two-level normal response variance component models was developed by Aitkin, Anderson and Hinde (1981), used in a ground-breaking reanalysis of a major educational survey in Aitkin, Bennett and Hesketh (1981), extended to two-level binary variance component models by Anderson and Aitkin (1985), and to generalized linear mixed models (GLMMs) by Anderson and Hinde (1988). Detailed applications and implementation in GLIM4 were discussed by Aitkin (1999a), based on earlier work on single-level random effect GLMs by Aitkin (1996) for which the EM algorithm was developed in GLIM4 by Aitkin and Francis (1995). The computational methods for model fitting in these papers were based on the EM algorithm, and are reviewed in Chapters 7–9 in Aitkin, Francis and Hinde (2005).

A version of the binary variance component algorithm was developed in a different context (binary test item data in large-scale psychometric and educational testing) by Bock and Aitkin (1981); it has been extended to more general forms of item response by many others (for example Masters 1982, and Mislevy and Bock 1986) and is the basis of much psychometric modelling, particularly in the NAEP. The item response model was not recognized as a variance component model until Adams, Wilson and Wu (1997).

The need for more than two levels in these models was recognized early in school comparison studies (Aitkin and Longford 1986) where students could be nested in classes nested in schools nested in Local Education Authorities.

Methods current in 1999 for fitting two-level models in the GLM family were reviewed by Aitkin (1999a). Few current proprietary statistical packages can handle nested structures for binary data with more than three levels; non-normal response models with Gaussian random effects are particularly difficult because of the non-analytic log-likelihood and the numerical integration steps needed at each random effect level; these are sometimes avoided by *quasi-likelihood* methods which do not use the actual likelihood and so do not provide a fully efficient ML analysis; their properties are generally inferior relative to those of full ML estimates.

Markov Chain Monte Carlo (MCMC) methods are now widely used for these models and can handle the non-conjugate Gaussian random effects; however as noted further below the EM algorithm-based maximum likelihood methods described here have a *major advantage* in being able to use *arbitrary* random effect distributions.

The proprietary statistical package Stata with the Gllamm add-on developed by Rabe-Hesketh, Skrondal and Pickles, at URL <http://www.gllamm.org/> can handle very general structures with an unlimited number of levels. However this program uses Gauss-Newton (GN) methods in which the first and second derivatives of the log-likelihood are computed numerically; this gives great generality but results in extraordinarily slow running on large-scale problems with high-order random effects – a NAEP data four-level model with 30 binary items with 10,763 cases, 60 item parameters, 24 explanatory variables and sparse item responses required 75 days of CPU time on a dedicated fast PC to achieve convergence in Gllamm.

Comparative timings of GLIM4 and Gllamm on the same data and computer with a two-level Rasch model, 30 item parameters and no explanatory variables gave 2.16 minutes for GLIM and 5 hours for Gllamm. For the two-level 2PL model with 60 item parameters and 24 variables, GLIM4 took 11 minutes, Gllamm 46 hours. The speed-up factor for GLIM4 was approximately 140 for the smaller model and 250 for the larger. If the larger model factor could apply to the four-level model, it would run in 7.2

hours, a practical time for routine data analysis. (Though GLIM4 is an older statistical package, it is one of the few in which the EM algorithm for two-level models has been implemented. In Latent Gold, described below, the speed-up factor for the 3-level EM algorithm implementation is even greater – more than 300.)

In the remainder of the paper we examine ways in which such a speed-up might be achieved.

### 3 The assessment of current packages

There are several currently available packages which can fit two or more hierarchical levels to binary item response data. The major packages we examined are described below.

#### 3.1 Gllamm

This is the only package currently available which can handle four levels with IRT models. We used it for all the analyzes of the simulated and the NAEP data on which we have reported.

Gllamm has been optimized by the Stata developers as far as possible. We had detailed discussions on Gllamm and Stata with Stata Corporation President Bill Gould and Chief Statistician Bobby Gutierrez at the October 2006 Australian and New Zealand Stata Users workshop in Melbourne. They said that the main reason for its slowness is the numerical integration: each hierarchical level multiplies the time by a factor equal to the number of quadrature points at that level; so for eight-point quadrature at each of three levels the time (relative to a single level model with no integration) is increased by a factor of 512. This however does not explain the slowness of the two-level model: this must be due to the numerical derivatives and/or the Gauss-Newton algorithm instead of the EM algorithm used in the GLIM program.

Speeding-up the Gllamm algorithm for multi-level IRT models would require a complete rewrite of the Stata code:

- allow the option of analytic derivatives wherever possible (for the call to the ML program in Stata);
- provide the option of using a hybrid EM/Gauss-Newton algorithm (as used in Aitkin and Aitkin 1996);
- parallelize the computationally intensive parts of the code using the StataMP (multi-processor) version.

#### 3.2 xtmixed in Stata

Stata has been substantially strengthened by the new Mata programming language. This is said to be almost as fast as C, but with safeguards C lacks. Gould claims that Stata's standard procedures (written in Mata) are now as fast as those of any competitor, and faster than most.

Stata's standard provision for multi-level models (excluding Gllamm) is the very fast `xtmixed` procedure, which is only for normal models, with any number of levels. A more general GLMM procedure is said to be under development; we have registered our interest to serve as a beta test site for the new procedure.

### 3.3 Latent Gold

This is a maximum likelihood package oriented to finite mixtures and latent variable models, particularly factor models. It allows very general models to be fitted, to the mixture multinomial probabilities as well as to the probability functions within each component. However it is unable to fit some simpler models needed for NAEP analysis, for example the 2PL model with a simple discrete latent ability distribution, for which it can fit only a mixture of normals distribution. A discrete (nonparametric) latent variable can be fitted with a Rasch model for two or three levels, but this might confound the latent ability with variations in the item discriminations.

Jeroen Vermunt has implemented the EM algorithm for three levels, so far only for Gaussian quadrature, based on his ingenious representation (Vermunt 2004) of the mixture probabilities resulting from the three levels of numerical integration, allowing efficient recursive computation of these probabilities needed in the M step of the EM algorithm. A user-defined number of random starting points is used to find the best local maximum of the likelihood to iterate. After the initial EM iterations, computation switches to the Gauss-Newton algorithm when near convergence to provide standard errors from the information matrix.

The package is very fast: Latent Gold is 300 times as fast as Gllamm on the 3-level 2PL NAEP analysis (1.75 hours on a small laptop compared with 522.6 hours on a faster cluster PC – nearly 22 days). Vermunt reports that he is revising the program syntax to allow more general models to be fitted, and is possibly extending the number of levels. The package is proprietary, so cannot be extended by the user at present.

Issues that need addressing for fitting NAEP-scale models are

- allowing program specification in scripts which can run in a Linux environment;
- memory management difficulties which limit the data and model size;
- allowing starting a model fit from a previously saved set of estimates;
- allowing nonparametric analysis with the 2PL model;
- extension to four levels;
- quadrature points for other distributions besides normal;
- extension to 3PL models;
- extension to MIMIC models.

However the speed on the 3-level model shows that the EM algorithm is both feasible and efficient for Gaussian quadrature with 2PL (not MIMIC) models for NAEP-scale data sets.

### 3.4 VARCL

VARCL was one of the first stand-alone maximum likelihood variance component modelling packages for normal data. It was developed as a menu-driven package for normal variance component and random coefficient models by Nick Longford in 1984-6. Longford implemented the efficient (scoring) algorithm in this model (LaMotte 1972, Longford 1989), which can be implemented for any number of levels by recursive computation of sums of squares and cross-products up and down the levels of the model. One of the versions of VARCL could handle up to nine levels of nesting.

The package is now distributed by Assessment Systems Corporation, at URL <http://www.assess.com/Software/VARCL.htm>

but in a very restricted form - only 24 parameters can be included in the regression model. VARCL has a quasi-likelihood analysis for generalized linear mixed models; there is no full ML analysis for these models.

### 3.5 MLWin

This is a well-established multi-level model package. It gives maximum likelihood estimates for normal hierarchical models with up to three nesting levels, but only quasi-likelihood estimates for other response distributions. These are inferior to full ML estimates from the numerical integration procedures in Gllamm and Latent Gold. Markov Chain Monte Carlo methods are also available in MLWin. These are discussed further below, under Bayesian methods.

### 3.6 HLM

This is a well-established multi-level model package. It gives maximum likelihood estimates for normal hierarchical models with up to three nesting levels, but only quasi-likelihood estimates for other response distributions (reference is made to the EM algorithm and the Laplace approximation), which are limited to two levels.

### 3.7 R

R is an open-source object-oriented statistical package (an open-source version of the proprietary S-plus package) which can be downloaded free from the various R mirror web sites, with a huge set of library functions. It has library multilevel functions `lme` and `lmer` for normal models, and there are several implementations in R of the GLIM4 macros by Aitkin and Francis for overdispersion and generalized linear mixed models, for example by Jochen Einbeck and John Hinde, at <http://www.nuigalway.ie/math/jh/npml.html>. The latter are restricted to two levels.

The R implementations are very memory-intensive and cannot fit any model more complex than the two-level Rasch model to the 1986 NAEP math data. For this model R took 25 minutes, compared with 2.16 minutes for the GLIM4 macro and 5 hours for Gllamm.

Since R is open-source, all its facilities are available for program development, including the vast libraries of R procedures. It is the development platform of choice for most statisticians, but is less suitable for large-scale data analysis as it runs very slowly

on large models and data sets, and can quickly run out of memory because of the accumulation of objects in the workspace. The R code for the implementations of the GLIM4 macros needs to be assessed to determine whether deletion of redundant objects would improve memory management and allow more complex models to be fitted. Also, intensive computation needs to be speeded-up by using “plug-in” C code.

Howard Doran at AIR and Doug Bates at Wisconsin are considering an extension of `lme` to generalized linear models.

### **3.8 SAS Proc NLMIXED**

This SAS procedure handles two sets of random effects in exponential family (and other) models by maximum likelihood using Gaussian quadrature, and so can handle three-level binary item data with person and school random effects, or two-level normal response data. Other non-linear (and non-GLM) models can be handled in the same framework. With more than two sets of random effects, the model has to be restructured to reduce the random effects to two levels, for example by marginalizing over one random effect level to give two sets of random effects, and an intra-class covariance matrix for the observed responses, conditional on the two sets of random effects.

### **3.9 WinBUGS**

This is a very widely used Bayesian package (the **Windows** version of **B**ayesian analysis **U**sing **G**ibbs **S**ampling) which is naturally oriented to hierarchical structures from its Bayesian formulation (priors for parameters are at a higher level than models for observed data). Further discussion of Bayesian methods is given below.

## **4 Extensions to the EM algorithm**

### **4.1 Extension of the EM algorithm for two-level GLMMs to four levels**

The EM algorithm works by treating the random effects as unobserved data, and replacing in the E step terms in the log-likelihood involving these effects by their conditional expectations given the observed data and the current parameter estimates.

With more than two levels, there are several sets of random effect terms, and in the non-normal response GLMMs their conditional expectations become progressively more complicated with the number of levels in the model.

In the two-level GLMM with only one set of Gaussian random effects at the upper level (the model used for binary item data with a normally distributed latent ability), these expectations require numerical integration over the discrete Gaussian masspoints, and the consequent M step can be expressed as a weighted version of ML with an additional masspoint factor and an expanded data set, with weights given by posterior probabilities of “membership” of each observation in the finite mixture “component” formed by each masspoint. These weights are calculated at the upper level but then applied to the ML analysis at the lower level (Aitkin 1999a).

A particular strength of the ML analysis, as noted above, is that it can handle, by a small extension, the case of *arbitrary* or *unspecified* random effects distributions, by estimating these distributions by nonparametric ML as discrete on a finite number of masspoints (Aitkin 1996, 1999a). The resulting estimate is equivalent to a *latent class model* for the random effects; by including interactions of the random effects with other explanatory variables, different regressions can be modeled in each latent class. This is generally not appropriate for ability but it may be very appropriate for school effects at a higher level. (If the random effects have a *specified* non-Gaussian distribution, for example  $t_4$ , masspoints and masses for this distribution simply replace those for the Gaussian distribution.)

This greatly increases the generality of the analysis, at the same time allowing a comparison of the Gaussian random effect model with others: a substantial improvement in the likelihood over the Gaussian distribution by estimating the random effect distribution (or by replacing it by another specified distribution) identifies failure of the Gaussian model to adequately represent the random effect variation. This feature of the EM analysis is much more difficult in MCMC because of the need for a random effect distribution prior on *the space of all distributions*; the Dirichlet process prior sometimes used (Escobar and West 1995) is particularly complicated (Aitkin 2001).

Aitkin (1996, 1999a, 1999b) gave several examples of the value of estimating the random effect distribution, and Aitkin, Francis and Hinde (2005 pp. 499-505) gave a detailed educational example. For NAEP this feature is of particular importance if percentile reporting is to continue.

Recently Vermunt (2004) has developed a recursive form for the EM algorithm in the multilevel GLMM which allows the weights to be computed cumulatively at each level and finally applied in the M step at the lowest level. This approach, which works for both Gaussian and unspecified random effects, provides the same level of generality as for the normal variance component model; the number of levels is (at least in theory) unlimited.

This algorithm has been implemented, though so far only for three levels and Gaussian quadrature, in Vermunt's proprietary package Latent Gold, distributed by Statistical Innovations, at the URL

[http://www.statisticalinnovations.com/products/latentgold\\_v4.html](http://www.statisticalinnovations.com/products/latentgold_v4.html).

## 4.2 The redesign of the M step of the algorithm to reduce the effective data size for this step

Both the single-level overdispersion and the two-level GLMM EM algorithms can be implemented (in GLIM and R) by *expanding the data size* by a factor equal to the number of quadrature masspoints  $K$ . This allows a simple weight specification to be applied to the expanded data set which weights together the SSP matrix terms calculated at each masspoint. If there are many quadrature points  $K$  this becomes a serious time and storage issue for large data sets – the effective data size is  $Kn$ ,  $K$  times the actual sample size  $n$ .

In GLIM the expanded data set does not have to be stored, but the actual data set still has to be looped through  $K$  times to calculate the weighted SSP matrix for the M step. This is wasteful, particularly for simple models with only one random

effect – overdispersion or simple variance component structure – because the explanatory variables are invariant over the masspoints and the same SSP matrix terms are computed at each masspoint: only the masspoint factor dummy variable is changed in the  $K$  loops through the data.

In more general models with random slopes as well as intercepts, the interaction terms also change over the masspoints, but their main effects do not, so again substantial savings are available by *not* repeatedly calculating the same terms at each masspoint.

With more levels, the current approach becomes unworkable because of the multiple expansions required – with two random effects each with 10 masspoints, the data are expanded by a factor of 100.

In the extension of the algorithm to more levels, full advantage will be taken of the constancy of explanatory variables across mass points, to reduce the effective data size. In particular, the SSP matrix for the “null” model without any random effects, which is currently used to provide initial parameter estimates for the algorithm, will also provide a substantial part of the full SSP matrix needed for the GLMM and will not be recalculated in the algorithm steps.

The extension will also compute at convergence the observed information matrix and standard errors for the fitted model by the Louis (1982) or Friedl-Kauermann (2000) methods: in the GLIM implementation standard errors are not provided for the parameter MLEs except by the device of dropping each individual variable after the full analysis and relating the change in deviance to the standard error (Aitkin, Francis and Hinde 2005 p. 373).

### **4.3 The parallelizing of the numerical integration step at each random-effect level**

The numerical integration step repeats the same computations at each of the  $K$  masspoints, as noted in 2 above. By assigning these computations to a set of  $K$  parallel processors, a considerable speed gain factor can be potentially achieved on this part of the computation, which is a major part of the full EM computation.

Developing the procedure to provide overnight run-times on large models will require several steps: a) recoding the algorithms in a compiled language, such as C, b) a parallel implementation, to distribute the work across many processors, and c) tuning the parallel implementation to approach linear speedup (if possible).

A summer internship student at VPAC (the Victorian Partnership for Advanced Computing) has implemented a prototype of phase a) and b), using the current EM algorithm in the R package. The correctness of both the compiled C library and its parallel version have been verified, but not yet checked for code efficiency. The speedup on small problems is counteracted by the overheads in message traffic to the multiple processors. Significant speedups should be achievable on large data sets, which cannot be fitted in the current R implementation.

Stata’s multiple processor version StataMP and the new programming language Mata provide the opportunity to check the feasibility and possible implementation of an efficient parallel program as above in Stata rather than R.

## 4.4 The acceleration of the M step by expanding the step-length

The EM algorithm is notoriously slow to converge with a high proportion of missing data, because of the replacement of the unobserved data by their conditional expectations, which are then treated as known data. This overstates the information in the data and leads to successive steps in the parameter space which are too small. Aitkin and Aitkin (1996) evaluated a hybrid EM/GN algorithm which used EM for the initial iterations and then switched to GN for faster convergence. On simple normal mixtures this saved some time, but only about 30% compared with pure EM iterations.

Jamshidian and Jennrich (1993) reformulated the EM steps as approximately generalized gradient steps and suggested that the algorithm be accelerated by generalized conjugate gradient methods which increase the step-length. They showed considerable improvements in the convergence rate of the resulting AEM (Accelerated EM) algorithm, especially in problems with large numbers of parameters and substantial proportions of “missingness” through latent variables. The time improvement factor ranged from 3 to 100 on different problems.

Lange (1995) and Jamshidian and Jennrich (1997) described alternative quasi-Newton acceleration methods which also change the step-length.

A new paper (Kuroda and Sakakihara 2006) gives a quite simple acceleration method for EM which does not require matrix inversion and has the same guaranteed convergence as EM. They report speed-up factors of 3-10 on different examples relative to EM.

The multilevel algorithm will be assessed for acceleration by the most appropriate of these improvements.

## 4.5 The inclusion of incomplete covariate data by computing a close approximation to the full information matrix for the incomplete data

In his PhD thesis under M. Aitkin’s supervision, Darnell (2003) examined several methods for including incomplete covariate data – observations with one or more covariate values missing randomly – into a full ML analysis for the normal response GLM. The common *complete case* analysis of incomplete data (by omitting the incomplete data) is widely recognized as inefficient and frequently biased; currently only multiple imputation methods allow properly for this inclusion (Rubin 1987, Schafer 1997), but these are not generally available for multilevel models.

Darnell (2003) examined the use of nonparametric maximum likelihood for the incomplete cases, treating them as finite mixtures over the missing covariates and estimating the covariate distribution nonparametrically by a finite mixture maximum likelihood EM algorithm. A similar approach was developed by Lawless, Kalbfleisch and Wild (1999), but this fails on continuous covariates with many values because the conditional distribution of the missing data given the observed data in general has no support points in the observed data with exactly the same values of the fully observed covariates. No software routines have been published for their approach, and a personal communication from Chris Wild suggests that any implementation will be restricted to discrete covariates with small numbers of categories. This is also the case for the approach which can

be implemented in Gllamm, of treating the missing data as latent variables (Aitkin and Scott 2006).

A detailed study by Aitkin and Chadwick (2003) for the US National Center for Education Statistics, comparing maximum likelihood and multiple imputation for incomplete data, suggested a composite approach which gave good results in normal GLMs with incomplete covariates, by obtaining parameter estimates using the EM algorithm assuming a multivariate normal distribution even for categorical covariates (as recommended by Schafer 1997 for multiple imputation), but computing the information matrix contributions for the incomplete cases by treating these cases as a finite mixture with mass  $1/m$  at each set of the relevant covariate values in the  $m$  complete cases (the empirical distribution of the covariates).

This method will be extended to general GLMs; simulations will assess the validity of the approximate methods and their possible extensions to multilevel models.

## 5 Bayesian methods

The use of Bayesian methods for the analysis of complex statistical models has increased dramatically in recent years, due partly to the increasing availability of cheap computing power, and partly to the extensive development of Markov Chain Monte Carlo (MCMC) methods (Gamerman 1997). The latter are notoriously computationally intensive, but for many complex non-linear models they are the only practical methods for analysis.

It is clear from the earlier discussions in this paper that MCMC methods are not *essential* for NAEP-scale analysis, but they may provide a *richer* and more *theoretically defensible* analysis than ML methods. These two advantages both derive from the dependence of ML methods on the validity of large-sample statistical theory, in which ML estimates are efficient and normally distributed with minimum variance. While some aspects of large-sample theory may be satisfactory for NAEP reporting (for example, the normality and efficiency of ML estimates of reporting group parameters), this may not be true for other aspects (like reporting group parameter SEs, and variance component estimates and SEs). Since standard errors are affected by the size of the variance components, their dependence on “plug-in” estimates of the variance components makes these SEs doubtful when the number of random effects is not large (for example at the PSU level), or when the variance component estimates themselves are small (as with the PSU level for the 1986 data).

The advantage of Bayesian methods here is that they do not require large samples for validity: the variability in parameter estimates is *correctly represented for any sample size* by very large-scale sampling directly from the full posterior distributions given by the computational output of the Markov Chain. These results also do not depend on informative or subjective prior distributions for the model parameters; standard non-informative priors are sufficient.

The main package used for MCMC is WinBUGS, mentioned above, or derivatives of it. Variance component models are easily specified in WinBUGS, and the normal multi-level model has a simple conjugate prior specification, with regression model parameters given diffuse normal priors and variance components given diffuse inverse gamma priors (or diffuse flat priors on the standard deviations).

The MCMC approach frequently uses *Gibbs sampling* in the chain computations.

Gibbs sampling works by alternating steps of the Data Augmentation Algorithm (Tanner and Wong 1987, Tanner 1996); these are similar to the E and M steps of the EM algorithm, but with full distributions of parameters and unobserved variables rather than MLEs and conditional means. For multi-level models, in the data step the random effects at each level are drawn randomly from their posterior distributions given the observed data and the parameters, and in the parameter step the parameters are drawn randomly from their posterior distributions given the observed data and the random effects. Alternating these steps leads to convergence of the parameter distributions to their joint posterior, and of the random effect distributions to their posterior distributions, given the observed data.

In the multilevel model with a normal response and normal random effects, all the posterior distributions above have simple analytic forms, and the simulations involved are very fast, though a large number of steps is needed for convergence - 50,000 is common. For the binary item outcomes at the bottom level, a probit model implies a normal latent threshold variable, and this also leads to a form of full normal analysis, affected only at the bottom level by the transformation from the latent threshold scale to the observed binary outcome scale. So a small change to the simple hierarchical normal model would allow the full analysis by Gibbs sampling. For the logit model an additional Metropolis-Hastings step would be needed as the logistic distribution is not conjugate with the rest of the normal model.

A detailed comparison of maximum likelihood and quasi-likelihood estimation with MCMC in 2- and 3-level normal and logistic regression models was carried out by Browne and Draper (2006). They found that in 2-level normal models, ML and REML-based methods were substantially faster than MCMC and had similar confidence interval coverage of both regression and variance component parameters. (REML – restricted maximum likelihood – methods maximize a marginal likelihood in which the regression parameters are integrated out of the likelihood, giving degrees-of-freedom corrections for the variance component parameters which have smaller biases, but generally larger MSEs, than the ML estimates for these parameters.)

In 3-level logistic regression models, full ML methods were not used in the study, but MQL or PQL (marginal or penalized quasi-likelihood) methods. The PQL or MQL estimates had poorer confidence interval coverage than MCMC for both regression and variance component parameters, and gave biased parameter estimates as well.

Full ML by Gaussian quadrature was not investigated, as being too slow computationally. An investigation of the relative performance of MCMC and full ML implementations of the four-level model would be of considerable value, as the MCMC output is much richer than just the MLEs and SEs of the model parameters – full information is available about the random effects as well, and posterior inference about these effects (usually done by plug-in empirical Bayes estimation) can be done with full accounting for the uncertainty in all the parameters. The computationally intensive parts of the MCMC algorithm may be also be parallelized, as for the EM algorithm, and this may substantially improve its performance.

## 6 Summary and developments proposed

It is clear from the above survey that routine analysis of NAEP-scale data, incorporating the full survey design, a large regression model incorporating relevant reporting group and other variables, and the psychometric models for test items, is feasible provided that dedicated special-purpose programs are developed for this analysis.

To develop such a program requires an extensive set of steps; not all of these can be completed in an 18-month project. The steps are not sequential; some will be done together:

- Implement the 2- and 3-level EM algorithms for 2PL models.
- Compare timings of different programs and platforms on NAEP-scale data.
- Compute the information matrix (for complete covariate data) for the 2- and 3-level EM algorithms.
- Implement the Gauss-Newton (GN) algorithm using the information matrix.
- Construct a composite algorithm with initial EM iterations switching to GN.
- Experiment with numbers of EM iterations for optimal convergence before switching to GN.
- Extend models to 3PL; experiment with methods to assess identifiability.
- Redesign the M-step to avoid data expansion.
- Experiment with step-length increments to accelerate convergence.
- Develop C code for computationally intensive parts of programs.
- Assess speed-up with parallel versions of the EM algorithm and MCMC, with varying numbers of processors.
- Develop methods for handling incomplete covariate data and the full information matrix.

## References

- Adams, R.J., Wilson, M. and Wu, M. (1997). Multilevel item response models: an approach to errors in variables regression. *Journal of Educational and Behavioral Statistics* **22**, 47-76.
- Aitkin, M. (1996) A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing* **6**, 251-262.
- Aitkin, M. (1999a) A general maximum likelihood analysis of variance components in generalized linear models. *Biometrics* **55**, 218-234.
- Aitkin, M. (1999b) Meta-analysis by random-effect modelling in generalized linear models. *Statistics in Medicine* **18**, 2343-2351.
- Aitkin, M. (2001) Likelihood and Bayesian analysis of mixtures. *Statistical Modelling* **1**, 287-304.
- Aitkin, M. and Aitkin, I. (1996) A hybrid EM/Gauss-Newton algorithm for maximum likelihood in mixture distributions. *Statistics and Computing* **6**, 127-130.
- Aitkin, M., Anderson, D.A. and Hinde, J.P. (1981) Statistical modelling of data on teaching styles (with Discussion). *Journal of the Royal Statistical Society A* **144**, 419-461.
- Aitkin, M., Bennett, S.N. and Hesketh, J. (1981) Teaching Styles and Pupil Progress: A re-analysis. *British Journal of Educational Psychology* **51**, 170-186.
- Aitkin, M. and Chadwick, T. (2003) *Full information from complex models – report on NCES/ESSI project 153*. National Center for Education Statistics, US Office of Education. Unpublished.
- Aitkin, M. and Francis, B.J. (1995) Fitting overdispersed generalized linear models by nonparametric maximum likelihood. *The GLIM Newsletter* **25**, 37-45.
- Aitkin, M., Francis, B.J. and Hinde, J.P. (2005) *Statistical Modelling in GLIM4*. Clarendon Press, Oxford.
- Aitkin, M. and Longford, N.T. (1986) Statistical modelling issues in school effectiveness studies (with Discussion). *J. Roy. Statist. Soc. A* **149**, 1-43.
- Aitkin, I. and Scott, J. (2006) The effect of missing data on covariates in survival analysis. Submitted to *Statistical Modelling*.
- Anderson, D.A. and Aitkin, M. (1985) Variance component models with binary response: interviewer variability. *Journal of the Royal Statistical Society B* **47**, 203-210.
- Anderson, D.A. and Hinde, J.P. (1988) Random effects in generalized linear models and the EM algorithm. *Communications in Statistics – Theory and Methods* **17**, 3847-3856.

- Bock, R.D. and Aitkin, M. (1981) Marginal maximum likelihood estimation of item parameters: an application of an EM algorithm. *Psychometrika* **46**, 443-459.
- Browne, W.J. and Draper, D. (2006) A comparison of Bayesian and likelihood-based methods for fitting multilevel models. *Bayesian Analysis* **1**, 473-514.
- Darnell, R.E. (2003) *General and approximate methods of maximum likelihood estimation with missing covariates in linear models*. PhD Thesis, University of Newcastle-upon-Tyne, UK.
- Dempster, A.P., Laird, N.M. and Rubin, D.A. (1977) Maximum likelihood estimation from incomplete data via the EM algorithm (with Discussion). *Journal of the Royal Statistical Society B* **39**, 1-38.
- Escobar, M.D. and West, M. (1995) Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* **90**, 577-588.
- Friedl, H. and Kauermann, G.(2000) Standard errors for EM estimates in generalized linear models with random effects. *Biometrics* **56**, 761-767.
- Gamerman, D. (1997) *Markov Chain Monte Carlo*. Chapman and Hall, London.
- Jamshidian, M. and Jennrich, R.I. (1993) Conjugate gradient acceleration of the EM algorithm. *Journal of the American Statistical Association* **88**, 221-228.
- Jamshidian, M. and Jennrich, R.I. (1997) Acceleration of the EM algorithm by using quasi-Newton methods. *Journal of the Royal Statistical Society B* **59**, 569-587.
- Kuroda, M. and Sakakihara, M. (2006) Accelerating the convergence of the EM algorithm using the vector  $\epsilon$  algorithm. *Comp. Statist. and Data Anal.* **51**, 1549-1561.
- LaMotte, L.R. (1972) Notes on the covariance matrix of a random, nested ANOVA model. *Annals of Mathematical Statistics* **43**, 659-662.
- Lange, K. (1995) A quasi-Newton acceleration of the EM algorithm. *Statistica Sinica* **5**, 1-18.
- Lawless, J.F., Kalbfleisch, J.D. and Wild, C.J. (1999) Semiparametric methods for response-selective and missing data problems in regression. *Journal of the Royal Statistical Society B* **61** 413-438.
- Longford, N.T. (1989) A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. *Biometrika* **74**, 817-827.
- Louis, T.A. (1982) Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society B* **44** 226-233.
- Masters, G.N. (1982) A Rasch model for partial credit scoring. *Psychometrika* **47**, 149-174.
- Mislevy, R., and Bock, R. D. (1986) *Bilog: Item analysis and test scoring with binary logistic models*. Scientific Software, Mooresville, IN.

- Rubin, D.B. (1987) *Multiple Imputation for Nonresponse in Surveys*. John Wiley, New York.
- Schafer, J.L. (1997) *Analysis of Incomplete Multivariate Data*. Chapman and Hall, London.
- Tanner, M.A. (1996) *Tools for Statistical Inference (3rd edn.)*. Springer, New York.
- Tanner, M.A. and Wong, W.H. (1987) The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association* **82**, 528-540.
- Vermunt, J.K. (2004) An EM algorithm for the estimation of parametric and nonparametric hierarchical nonlinear models. *Statistica Neerlandica* **58**, 220-233.