ED 340 731 TM 017 778

AUTHOR	Kreft, Ita G. G.; And Others
TITLE	Comparing Four Different Statistical Packages for Hierarchical Linear Regression: GENMOD, HLM, ML2, and VARCL.
INSTITUTION	Center for Research on Evaluation, Standards, and Student Testing, Los Angeles, CA.
SPONS AGENCY	Institute for Educational Research in the Netherlands (SVO), The Hague.
REPORT NO	CSE-TR-311
PUB DATE	Feb 90
NOTE	112p.
AVAILABLE FROM	CSE Dissemination Office, UCLA Graduate School of Education, 405 Hilgard Avenue, Los Angeles, CA 90024-1521.
PUB TYPE	Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC05 Plus Postage.
DESCRIPTORS	Comparative Analysis; *Computer Software; Computer Software Evaluation; Educational Assessment; Elementary Education; Equations (Mathematics); Foreign Countries; *Mathematical Models; *Regression (Statistics)
IDENTIFIERS	GENMOD Computer Program; *Hierarchical Linear Modeling; HLM Computer Program; ML2 Computer Program; Nested Data; Netherlands; VARCL Computer Program

ABSTRACT

An overview is given of the available statistical theory and software for analyzing hierarchically nested data. Programs are evaluated, and general techniques are proposed to analyze data from several domains. This research is part of a larger project to evaluate elementary education in the Netherlands. The models discussed are the random coefficient models, the hierarchical mixed linear models, and the multilevel linear models. The abstract characteristics of the three classes of models and the systematic treatment of random and non-random parts of each class are described. Transformation of the models and the likelihood function are considered. The following four computer programs, using various types of algorithms, are discussed: (1) GENMOD; (2) HLM; (3) ML2; and (4) VARCL. Each is compared for design, implementation, performance and results, and ease of use. To overcome some of the disadvantages of these techniques, a new program, MULTIPATH, is proposed for a more general approach to the analysis of data from different domains. Thirteen data tables and a 61-item list of references are included. (SLD)

TM

COMPARING FOUR DIFFERENT STATISTICAL PACKAGES FOR HIERARCHICAL LINEAR REGRESSION GENMOD, HLM, ML2, AND VARCL

CSE Technical Report 311

Ita G.G. Kreft
Jan de Leeuw
Kyung-Sung Kim

UCLA Center for Research on Evaluation,
Standards, and Student Testing

February, 1990

The research reported herein was conducted witl. partial support from the Institute for Educational Research in the Netherlands under grant 7130. However, the opinions expressed do not necessarily reflect the position or policy of this agency and no official endorsement by this agency should be inferred.

Version 3.0 of Report #1 of SVO-project 7130: "Developments of Computer Programs for Multilevel Analysis in Educational Evaluation." The authors thank Bill Oldendorf and Rebecca Frazier for their assistance in editing this report, and Steven Raudenbush, Tony Bryk, Nick Longford, Bill Mason, Harvey Goldstein, and Bob Prosser for their extensive and invaluable comments on Versions 1.0 and 2.0 (February 1990).

Please address inquiries to: CSE Dissemination Office, UCLA Graduate School of Education, 405 Hilgard Avenue, Los Angeles, California, 90024-1521

Abstract

"...Then anyone who leaves behind him a written manual, and likewise anyone who receives it, in the belief that such writing will be clear and certain, must be exceedingly simple-minded..."

Plato, Phaedrus 275d

This research is part of the larger project "The Evaluation of Primary Education in The Netherlands". In this project the following research questions are studied:

(1) Are significant positive effects produced by the change from primary education for 6 to 12 year olds, to basic education (*basisvorming*) for 4 to 12 year olds ?

(2) Are there measurable differences between, on the one hand, schools with and without *basisvorming* and, on the other, schools with experimental basic education ?

(3) Do structural and organizational differences of basic schools produce distinguishable effects ?

All three research questions involve data measured at at least two levels. For the first two questions these are the student and the school levels, for the last question we also have the regional or structural level as a third level. Because of this hierarchical structure special demands are placed upon the analysis technique which can be used to answer these questions.

In this report we give an overview of some of the available statistical theory and software for analyzing hierarchically nested data. Next, we evaluate these programs. Finally, we propose a technique that can analyze data of several different domains and is (at least in some respects) more general than the existing programs.

1. INTRODUCTION.

1.1. Why new techniques ?

Evaluating the effectiveness of large-scale experiments in education involves the analysis of *hierarchical data structures*. Educational data are often hierarchical because pupils are in schools, schools are in districts, districts are in counties, and counties are in states. In a large-scale research project we usually have information about two or more of the levels involved, for instance variables describing individuals (such as intelligence, school career, and family background), variables describing the schools (school type, schools in a special program, curricula offered), and perhaps variables describing districts or countries (available resources). It is well known that analysis of these variables on any of these levels separately can be seriously misleading. For an overview see Burstein, 1980, and Kreft, 1987. It is more satisfactory to construct models and techniques which simultaneously take information of all levels into account. But in order to be able to do this some serious statistical problems have to be solved. Problems in hardware and software that were unsolvable until recently. In the last few years, however, a number of papers in the statistical and methodological literature directly have attacked the problem of analyzing variables measured at different levels of a hierarchy. (See Mason, Wong, & Entwistle, 1985; Aitkin & Longford, 1986; Goldstein, 1986; Raudenbush & Bryk, 1986; and De Leeuw & Kreft, 1986.) These investigators work with basically the same model, known as the *hierarchical linear model*, the *random coefficient model*, or the *Bayesian linear model*. All models deal with the problem of analyzing nested data collected under non-experimental conditions.

Mathematical models can serve the goal of describing relationships between variables of different levels. Use of a mathematical model indicates the researcher's assumption that the structure of the model is more or less in correspondence with that of the real world. The model describes the empirical world as a formalized theory. Preference for one model over another is based on the researcher's theoretical knowledge of the subject and knowledge of the way the data are produced.

For instance the hierarchical linear model, the model we are discussing here, is preferable to traditional models such as multiple regression and analysis of covariance (ANCOVA), when the assumptions of the latter are violated. As a result of this violation we assume that estimators based on the more realistic model are less biased and more efficient. Let us give an example with educational data, comparing the hierarchical model with ANCOVA, a traditional fixed linear model. In ANCOVA the most obvious assumption violated is that of random sampling. This is clear, since in education we are dealing with data in which students are sampled from within schools and more often than not even these schools are not a random sample (but are stratified or otherwise sampled). Moreover, school populations do not consist of a random sample of the country's population of students, but are highly selective, causing a restricted range of many important variables. Selection often takes place on the dependent variable school success, since schools are in poor or rich neighborhoods, and the relation between school success and social economic status is obvious in many instances. In addition, when we want to apply within-school analysis, the samples within schools are not large enough to make up for this non-normality of the data. As a result, the ordinary least squares (OLS) estimates of the traditional linear model provide confidence intervals that are too short and too often the null hypotheses will be rejected. This causes many type I errors, and as a result contradictory research results.

The literature tells us that the choice of the best model depends on the simplest description of the population studied that is consistent with the data (see Everett & Dunn, 1983, for instance). Of course, within certain limits, it is always true that when we make our models complex enough, they are bound to fit. But a complicated model may have less explanatory power than a simple and more elegant one; a simpler model also may be easier to interpret. This is the trade-off between parsimony and goodness of fit. In April 1987, at E.T.S. in Princeton, there was a conference especially devoted to studying problems of this kind. The proceedings were published in a very useful book (Bock, 1988). The book, and many recent articles, show that various groups of researchers are still busy extending the theory as well as the existing models and techniques for analyzing this type of data. We also see the trend of extending these techniques to more general types of situations, and applying them to domains other than school effectiveness research (see Laird & Ware, 1982; Jennrich & Schluchter, 1986; Hox, Kreft & Hermkes, 1988).

The question we study in detail is how to fit separate models for separate schools in an elegant and parsimonious way. In order to do this we present a detailed comparison of four computer programs for analyzing hierarchical linear models. The programs we discuss are VARCL, HLM, ML2, and GENMOD. We have selected them on the basis of the following criteria. We restrict our attention to programs that are (a) compiled, (b) stand-alone, and (c) specialized. On the basis of (a) and (b) we exclude the GGCMAOV routine by Stram, Laird and Ware (1986), which is written in SAS/MACRO and in SAS/MATRIX, and the routine by Kim (Kim & Kreft, 1989), which is written in GAUSS. Because of (c) we exclude general purpose mixed-model ANOVA routines, and because of (b) we do not discuss packaged modules such as BMDP/5V (Schluchter, 1988) or GLM and VARCOMP from the SAS statistics package. The BMDP/5V routines of Schluchter, and GGCMAOV of Stram, Laird, and Ware, were written with longitudinal (growth curve, repeated measures) data in mind. They will be discussed and compared in a separate paper (Van der Leeden & De Leeuw, 1990).

The four programs produce their own answers, but basically their answers lead to the same conclusion. OLS produces biased and unstable estimates of the parameters and their precision, since the assumptions on which OLS is based are clearly violated. The specialized methods can produce more stable estimates. There are some quite subtle differences among the models that can be fitted by the programs, but in general these approaches have a great deal in common. Most programs concentrate on two parts, or *levels* to decompose, although having more than two is possible (HLM 3 and ML 3 have been recently released in beta-form; VARCL already exists as VARCL 3 and VARCL 9). Using an educational example again, these levels may be the variance between students and the variance among schools. In this example, students are the first level and schools the second, but the levels could be defined differently. All four programs use *Maximum Likelihood* (ML) estimation for decomposition of the variance into different parts. In all cases computing the ML estimates involves complex nonlinear expressions in the parameters. In such cases the equations must be solved by an iterative procedure. The major difference among the programs is in the choice of the criterion to optimize and the choice of the algorithm to optimize the criterion.

1.2. Models.

There has been quite a bit of confusion in the literature about the precise differences among the programs, the distinction between restricted and full ML, the precise definition of the algorithm, and the various reasons one can use to choose among programs and options. This may be due to the fact that the investigators have concentrated on making the software work and publishing the application-oriented papers. It also may be due to the fact that some things are taken for granted, which are not obvious at all. Why do we use ML? Why do we choose restricted ML rather than full ML? What is the EM algorithm? We will try to give a clear outline of the various tools that are used and of the various aspects of the packages.

Our first step is to suggest a uniform terminology and notation. Although, of course, notation and terminology should not be a barrier to understanding other people's work, they often have this effect. We then introduce the models at three different levels of generality: (a) the most general models, the *Random Coefficient Models*; (b) an important special case, the *Hierarchical Mixed Linear Models*; and (c) the *Multilevel Linear Models*, which are the most specialized models, although they are still a very general class. We will discuss most problems at their appropriate level of generality. For example, in the random coefficient model context, for instance, there is a minimum of notation and various technical matters can be discussed by using matrices, without cluttering the page with a primeval forest of formulas. Simplifications are possible for the more specialized models, and this leads to some unpleasant algebra, but the equations will show where computational efficiency can be found.

In general we will emphasize the abstract characteristics of the three classes of models and the symmetric treatment of random and nonrandom parts of each class. In this context it is perhaps necessary to stress a philosophical point. What we model as "random" and what we model as "fixed" should be ultimately decided by a *replication* of the experiment or study. If our definition of a replication implies that certain variables will have the same values over different replications, then they will be modeled as fixed. If variables are allowed to change over replications, then they will be modeled as random. Another philosophical point is that we do not stress the distinction between variables and parameters very much, only that between variables which are observed and variables which are hypothetical and not observed. This could be considered as a small step in the direction of Bayesian statistics, but there is nothing inherently Bayesian about our interpretations.

1.2.1 Random Coefficient Models

In order to fix the notation for random coefficient models we discuss a completely general case in which we observe measurements of n *individuals* on a single *outcome* variable y. The outcome variable is often called the *dependent variable*, but this is quite misleading. The observations y_i are interpreted as realizations of a random variable $\underline{y}_i$. We shall underline random variables in our discussion of models. This convention (Hemelrijk, 1966) is especially compelling in our context, in which the distinction between fixed and random is of major importance. For each individual we also have observations x_{ij} on m *predictors* x_j. The x_j are thought of as fixed (by design). They are often called *independent variables* or *regressors*. Observe that, by definition, random variables are unobserved. We observe only a single realization of them in the particular trial of the experiment or study at hand. Or, to put it differently, random variables enter only into the formulation of the models; they do not figure in the formulas for the maximum likelihood loss functions or in the notation for observed values.

Our model for the $\underline{y}_i$ supposes that for each individual there exists an m-element vector of *random coefficients* $\underline{\beta}_i$ and a one-dimensional *residual* $\underline{\varepsilon}_i$ such that

$$(1):\ \underline{y}_i = \sum_{j=1}^{m} x_{ij}\underline{\beta}_{ij} + \underline{\varepsilon}_i$$

with $E(\underline{\varepsilon}_i) = 0$ and $V(\underline{\varepsilon}_i) = \sigma^2$ for all i. Moreover all $\underline{\varepsilon}_i$ are independent of one another and of all β_{ij}. The basic assumption on residuals is that there is no structure left in them—they are completely random.

Now let

(2a): $E(\underline{\beta}_{ij}) = \beta_{ij}$

(2b): $\underline{\beta}_{ij} - \beta_{ij} = \underline{\delta}_{ij}$

This means that

(3): $\underline{y}_i = \sum_{j=1}^{m} x_{ij}\beta_{ij} + \sum_{j=1}^{m} x_{ij}\underline{\delta}_{ij} + \underline{\varepsilon}_i$

and $E(\underline{\delta}_{ij}) = 0$. The first summation in (3) defines the *fixed part* of the model, the second summation the *random part*. It follows from the assumptions so far that expectation and covariance of the outcomes are given by

(4a): $E(\underline{y}_i) = \sum_{j=1}^{m} x_{ij}\beta_{ij}$

(4b): $C(\underline{y}_i, \underline{y}_k) = x_i^T \Omega_{ik} x_k + \sigma^2 \delta^{ik}$

where

(4c): $\Omega_{ik} = E(\underline{\delta}_i \underline{\delta}_k^T)$

and δ^{ik} is the Kronecker delta (i.e. $\delta^{ik} = 1$ if $i = k$, and $\delta^{ik} = 0$ otherwise). This model is very simple, very general, and perfectly useless. With each single observation y_i we introduce m fixed parameters β_i. With each pair of observations we introduce m(m+1)/2 parameters Ω_{ik}. Moreover there is the additional parameter σ^2. In general there will be far too many parameters to estimate, and far too few observations on which to base the estimation.

Therefore we must impose *restrictions* on the parameters, or, in other words, we must make our models more specific. One typical set of restrictions defines the *usual linear model*, in which there is no nontrivial random part and the fixed coefficients are all equal. Thus $\Omega_{ik} = 0$ and $\beta_{ij} = \beta_j$. The model becomes $E(\underline{y}) = X\beta$ and $V(\underline{y}) = \sigma^2 I$. In this model m + 1 parameters are left to estimate, and we all know how to estimate them by using ordinary least squares. A different set of restrictions defines the *variance components model*, which does not have a fixed part and which has all Ω_{ik} equal and diagonal. Thus, $\beta_{ij} = 0$ and $\Omega_{ik} = \text{diag}(\Omega)$, and the model is $E(y) = 0$, $V(y) = X\Omega X^T + \sigma^2 I$. Again there are m + 1 free parameters, but they are not used now to model the expected values of the outcomes; rather, they are used to model their variances and covariances. Both sets of restrictions are very strong and, moreover, they do not take any hierarchical structure of the data into account. In order to model this structure, we first have to define it precisely.

1.2.2 Hierarchical Mixed Linear Models

We start with the *index set* $N = \{1,2,...,n\}$. Suppose $\Pi_1,\ldots,\Pi_s$ are *nested partitionings* of N, i.e. all Π_r are partitionings of N, and the sets in Π_{r+1} are subsets of the sets in Π_r. We can also say that Π_{r+1} is a *refinement* of Π_r. One special partitioning is $\Pi_{min} = \{\{1,2,...,n\}\}$, i.e. Π_{min} consists of one subset, all of N. Another one is $\Pi_{max} = \{\{1\},\{2\},...,\{n\}\}$, i.e. Π_{max} consists of all singletons from N. For interpretation purposes, we can think of the elements of N as students, for instance. Π_1 partitions the whole set of students (in the investigation) into countries, Π_2 partitions countries (and thus students) into cities, Π_3 partitions cities (and thus countries and thus students) into neighborhoods, Π_4 partitions neighborhoods into schools, Π_5 partitions schools into classes, and Π_6 partitions classes into individual students. This is the finest partitioning. We indicate the nestedness of a given *hierarchy* $\{\Pi_r\}$ by writing $\Pi_{min} \leq \Pi_1 < \ldots < \Pi_s \leq \Pi_{max}$. The numbers $1 \leq n_1 < \ldots < n_s \leq n$ indicate the number of sets in each of the partitionings. Thus n_r is the number of sets in Π_r. With each partitioning we associate an *equivalence relation*. Suppose $=_r$ is the equivalence relation defined by Π_r. Thus $i =_r k$ if and only if individuals i and k are in the same subset on level r. Students are equal at the school level if they are in the same school, they are equal at the county level if their schools are in the same county, and so on. We have $i =_{min} k$ for all i and k, and $i =_{max} k$ if and only if $i = k$.

We now assign to each variable its *level* in the following way. Let us call a predictor x_j *of level r*, with $1 \leq r \leq n$, if

$$(5)\quad (i =_r k) \Rightarrow (x_{ij} = x_{kj})$$

If x_j is of level r, then it also is of level t with $t > r$. A variable of level r can have at most n_r different values Thus a predictor of level one is a necessarily a constant, a predictor of level n *can* have all its values different. Observe, however, that a constant *can* also be a predictor of level r, with $r > 1$. The definition extends, without problems, to random variables. We say that variable $\underline{\beta}_j$ has level r if

$$(6)\ (i =_r k) \Rightarrow (\underline{\beta}_{ij} \equiv \underline{\beta}_{kj}),$$

where we use ≡ for equivalence of random variables (that is, equality up to sets of probability zero). Given these definitions, it is now clear what we mean by a *hierarchical data structure.* It consists of a number of variables on N, plus a nested set $\{\Pi_r\}$ of partitionings of the individuals in N. It is also clear what we mean by a *multilevel data structure*. It is a hierarchical data structure, with variables (defined on N, fixed and/or random) of different levels.

We now incorporate the hierarchical structure of the data into the general model (3). Suppose that coefficients $\underline{\delta}_{ij}$ are partitioned into subsets as

$$(7):\ \underline{\delta}_{ij} = \left(\underline{\delta}_{ij}^{(1)} \mid \dots \mid \underline{\delta}_{ij}^{(s)}\right)$$

where subset r has m_r elements of level r. Thus there are disturbances on all levels of the hierarchy. We now link the hierarchical structure of the data set and the correlation of the error terms by the following assumption

$$(8a):\ \text{if } r \neq t \text{ then } E\left(\underline{\delta}_{ij}^{(r)}\underline{\delta}_{kl}^{(t)}\right) = 0$$

$$(8b):\ \text{if } i \neq_r k \text{ then } E\left(\underline{\delta}_{ij}^{(r)}\underline{\delta}_{kl}^{(r)}\right) = 0$$

$$(8c):\ \text{if } i =_r k \text{ then } E\left(\underline{\delta}_{ij}^{(r)}\underline{\delta}_{kl}^{(r)}\right) = \omega_{jl}^{(r)}$$

This means that all between-level correlations are zero. If individuals i and k are not in the same equivalence class on level r, then their level-r random coefficients are uncorrelated. If they are in the same equivalence class, then their level-r random coefficients are by definition identical because they are of level r. Moreover, the blocks of covariances for all equivalence classes are equal.

We partition the columns of X in exactly the same way as the random coefficients. Thus

$$(9):\ X = \left(X^{(1)} \mid \dots \mid X^{(s)}\right)$$

with no restrictions on the level of $X^{(r)}$. We also assume that the fixed coefficients are the same for all i. Then

$$(10):\ \underline{y}_i = \sum_{r=1}^{s}\sum_{j=1}^{m_r} x_{ij}^{(r)}\beta_j^{(r)} + \sum_{r=1}^{s}\sum_{j=1}^{m_r} x_{ij}^{(r)}\underline{\delta}_{ij}^{(r)} + \underline{\varepsilon}_i$$

It follows that

$$(11a):\ E(\underline{y}_i) = \sum_{r=1}^{s} \sum_{j=1}^{m_r} x_{ij}^{(r)} \beta_j^{(r)}$$

$$(11b):\ C(\underline{y}_i, \underline{y}_k) = \sum_{r=1}^{s} \delta_r(i,k) \sum_{j=1}^{m_r} \sum_{l=1}^{m_r} x_{ij}^{(r)} x_{kl}^{(r)} \omega_{jl}^{(r)} + \sigma^2 \delta^{ik}$$

where $\delta_r(i,k) = 1$ if $i =_r k$ and $\delta_r(i,k) = 0$ otherwise. Observe that if $\delta_r(i,k) = 1$, then $\delta_t(i,k) = 1$ for all $t < r$ as well. Clearly (11a) and (11b) specialize (4a) and (4b). Let us translate (11) into matrix notation. It then says that

$$(12a):\ E(\underline{y}) = \sum_{r=1}^{s} X_r \beta_r$$

$$(12b):\ V(\underline{y}) = \sum_{r=1}^{s} \Delta_r\left[X_r \Omega_r X_r^T\right] + \sigma^2 I$$

where $\Delta_r[.]$ multiplies its argument elementwise by $\delta_r(i,k)$ (i.e., it zeroes out the covariances between nonequivalent individuals on each level).

Additional restrictions can be imposed that require either some of the β_{rj} to be zero (a variable does not occur in the fixed part) some of the rows and columns of the Ω_r to be zero (a variable does not occur in the random part). Model search usually takes place in the general class of models defined by (11a) and (11b), over the submodels specified by the various zeroes (or otherwise fixed numbers).

1.2.3. Multilevel Models.

We now define multilevel models in a quite general way. Suppose we have a hierarchy $\{\Pi_r\}$. The outcome variable $\underline{y}$ is of the most refined level s. In fact we assume that $s = n$ (i.e., none of the $\underline{y}_i$ are *necessarily* equal, though they could very well be *accidentally* equal—think of the case in which the outcome is binary, as in passing or not passing an exam). For each level r we have m_r predictors of level r, all defined on N. They are collected in the matrices $X^{(r)}$. Our first equation is

$$(13):\ \underline{y}_i = \sum_{j=1}^{m_s} x_{ij}^{(s)} \underline{\beta}_{ij}^{(s-1)} + \underline{\varepsilon}_i^{(s)}$$

Thus, for example, columns of $X^{(s)}$ are predictors on the student level (remember this does not exclude that there are school variables, county variables, and even constants among them), and the columns of $\underline{B}^{(s-1)}$ are random coefficients on the school level (that is, if students i and k are in the same school then their betas are equal). The disturbances $\underline{\varepsilon}^{(s)}$ in (13) have the usual structure. They are independent, identically distributed, centered, and they are independent of the random coefficients.

Now take the second step in the specification. For the elements of $\underline{B}^{(s-1)}$ we assume a model of the form

$$(14):\ \underline{\beta}_{ij}^{(s-1)} = \sum_{l=1}^{m_{s-1}} x_{ijl}^{(s-1)} \underline{\beta}_{ijl}^{(s-2)} + \underline{\varepsilon}_{ij}^{(s-1)}$$

Here the random regression coefficients of level s - 1 are linear combinations of fixed predictors of level s - 1 and new random regression coefficients of level s - 2 (plus a disturbance term of level s - 1). Observe that $X^{(s-1)}$ and $\underline{B}^{(s-2)}$ both have m_{s-1} columns. Disturbances $\underline{\varepsilon}^{(s-1)}$ are independent of previous disturbances and of disturbances of the level s - 1 that correspond with different equivalence classes. They are identically distributed over equivalence classes. Of course the disturbances have zero expectations: We do not assume that they are independent within equivalence classes (for different variables j and l).

Progression to the next level is now clear. On level s (which is level n) we have nm_s random coefficients, which form m_s variables of level s - 1. Thus, they take at most $m_s r_{s-1}$ different values. On the second level we have $nm_s m_{s-1}$ random coefficients, forming $m_s m_{s-1}$ variables of level s - 2, that is takes at most $m_s m_{s-1} r_{s-2}$ values. And so on. To stop the sequence starting at (13) and (14), we usually assume that coefficients on the last (crudest) level are no longer random (have zero disturbances).

In order to indicate how multilevel models are a special case of hierarchical mixed linear models we substitute (14) into (13). This gives

$$(15):\ \underline{y}_i = \sum_{j=1}^{m_s} x_{ij}^{(s)} \left\{ \sum_{l=1}^{m_{s-1}} x_{ijl}^{(s-1)} \underline{\beta}_{ijl}^{(s-2)} + \underline{\varepsilon}_{ij}^{(s-1)} \right\} + \underline{\varepsilon}_i^{(s)} = \sum_{j=1}^{m_s} \sum_{l=1}^{m_{s-1}} x_{ij}^{(s)} x_{ijl}^{(s-1)} \underline{\beta}_{ijl}^{(s-2)} + \sum_{j=1}^{m_s} x_{ij}^{(s)} \underline{\varepsilon}_{ij}^{(s-1)} + \underline{\varepsilon}_i^{(s)}$$

It is clear how this substitution continues when more levels are involved. If level (s - 1) is the highest level we are interested in, then we stop our recursion by assuming that $\underline{B}^{(s-2)}$ is fixed and of minimum level. Thus all its elements are equal to the fixed parameters β_{jl} (which are the same for all students). This gives

$$(16):\ \underline{y}_i = \sum_{j=1}^{m_s} \sum_{l=1}^{m_{s-1}} x_{ij}^{(s)} x_{ijl}^{(s-1)} \beta_{jl} + \sum_{j=1}^{m_s} x_{ij}^{(s)} \underline{\varepsilon}_{ij}^{(s-1)} + \underline{\varepsilon}_i^{(s)}$$

Now (16) gives a model in which the fixed part consists of products of variables of different levels (interactions). If the models (13) and (14) have constant terms, then some of the products will degenerate to some of the variables themselves. The random part of the model also has product variables, but of lower order. In the case of (16), no actual products occur in the random part of the formulation because we only have two levels.

Once again we take expectations to see how (16) specializes (4) and (11). Before we do this, we extend the model (as an exercise) to three levels. We have to add the specification

$$(17):\ \underline{\beta}_{ijl}^{(s-2)} = \sum_{v=1}^{m_{s-2}} x_{ijlv}^{(s-2)} \underline{\beta}_{ijlv}^{(s-3)} + \underline{\varepsilon}_{ijl}^{(s-2)}$$

Substitute this into (15). Then

$$(18):\ \underline{y}_i = \sum_{j=1}^{m_s} \sum_{l=1}^{m_{s-1}} \sum_{v=1}^{m_{s-2}} x_{ij}^{(s)} x_{ijl}^{(s-1)} x_{ijlv}^{(s-2)} \underline{\beta}_{ijlv}^{(s-3)} + \sum_{j=1}^{m_s} \sum_{l=1}^{m_{s-1}} x_{ij}^{(s)} x_{ijl}^{(s-1)} \underline{\varepsilon}_{ijl}^{(s-2)} + \sum_{j=1}^{m_s} x_{ij}^{(s)} \underline{\varepsilon}_{ij}^{(s-1)} + \underline{\varepsilon}_i^{(s)}$$

Now make the usual constancy assumption for the random coefficients on the lowest level. Taking expectations then gives

$$(19a):\ E(\underline{y}_i) = \sum_{j=1}^{m_s} \sum_{l=1}^{m_{s-1}} \sum_{v=1}^{m_{s-2}} x_{ij}^{(s)} x_{ijl}^{(s-1)} x_{ijlv}^{(s-2)} \beta_{jlv}$$

$$(19b):\ C(\underline{y}_i, \underline{y}_k) = \sum_{j=1}^{m_s} \sum_{l=1}^{m_s} \sum_{u=1}^{m_{s-1}} \sum_{v=1}^{m_{s-1}} x_{ij}^{(s)} x_{ijv}^{(s-1)} x_{kl}^{(s)} x_{klu}^{(s-1)} \delta_{s-2}(i,k) \omega_{jluv}^{(s-2)} + \sum_{j=1}^{m_s} \sum_{l=1}^{m_s} x_{ij}^{(s)} x_{kl}^{(s)} \delta_{s-1}(i,k) \omega_{jl}^{(s-1)} + \sigma^2 \delta^{ik}$$

Formula (16) is clearly equivalent to (11) in the special case of only one level (schools) besides the highest one (students). Fonnula (19) is (11) for the case of three levels. In multilevel models the special product structure defining the interactive variables is added. For more than two or three levels the notation becomes embarrassing. It is much better to discuss the general case in the context of the hierarchical mixed linear model.

1.2.4 Transformation of Models

In our discussion of transformations we use the notion of *invariance*. This means (roughly) that if we transform the variables in the model by any *admissible* transformation, then the results given by the technique should not change in any essential way.

In the linear model we often think that the results of the analysis should be invariant over all linear combinations of the predictors. In mixed linear models (with both fixed and random parts) we often require invariance only over linear combinations of the predictors corresponding with fixed coefficients. In the random coefficient and hierarchical models the situation becomes somewhat more complicated, because the same predictors occur in both the fixed and random parts. This has led to some confusion (Raudenbusch, 1989a, 1989b; Longford, 1989a; Plewis, 1989).

Suppose, in (12) for instance, that the admissible transformations of X_r are of the form X_rT, where $T \varepsilon \mathcal{T}$, a class of singular matrices. If we transform the regression coefficients to $\underline{b}_r = T^{-1}b_r$ and the dispersions to $\underline{\Omega}_r = T^{-1}\Omega_r(T^{-1})^T$, then we fit exactly the same model. The only thing we have to be careful about is that if we require certain elements of b_r and Ω_r to be zero, then $\underline{b}_r$ and $\underline{\Omega}_r$ should have the same pattern of zeroes for all T in $\mathcal{T}$, otherwise the pattern of zeroes is not an invariant. This is especially important in the case in which the first column of the X_r has all elements equal to +1 (it is the *intercept* of the regression). Centering of some or all of the remaining columns in X_r then is of the required form X_rT, and for invariance we need to assume that elements in the first row and column of Ω_r are not restricted to zero or to other fixed constants.

Observe that if there is no intercept in the regression, then centering cannot be written as

X_rT, but it must be written as JX_r, with $J = I - n^{-1}uu^T$, where u has elements equal to +1. If we interpret y and X as their centered versions, then (12) remains true, except for one minor detail. In (12b) we must replace σ^2I by $\sigma^2J = \sigma^2I - \sigma^2uu^T$. This trivial modification means that we do *not* have exact invariance in this case. The situation becomes even more complicated if we use premultiplication by a general projector P (for instance if the columns of PX_r are deviations from subgroup means). Again this is similar to fitting the same model on the Py, except for the disturbance term σ^2P. Also observe that even this restricted form of invariance applies only if we premultiply all columns in all X_r by P, irrespective of their level.

For multilevel models such as (16) or (18) the situation becomes more involved because of the interaction variables. If we make a linear transformation a + bx of a variable x, then this linear transformation will affect all products in which the variable x occurs. If we transform x to a + bx and z to c + dz, then xz is transformed to ac + bcx + adz + bdxz, and thus we only get invariance in the fixed part if the original model had xz, but also x, z, and an intercept. In general, it is clear that the consequences of transformations such as centering or centering around group means can be traced fairly easily using simple algebra. If one insists on a particular form of invariance, then this means that some variables have to be present and some coefficients cannot be restricted to be equal to zero. It is difficult to give clear-cut rules here, except for the obvious ones we have already discussed.

1.3 The Likelihood Function

1.3.1. Full Information Likelihood Function.

Even our most general class of models, the random coefficient models, are special cases of the *general mixed linear model*, and the general mixed linear model is a special case of the *heteroscedastic linear model.* For computational purposes, let us look at the heteroscedastic linear model first.

In the general heteroscedastic model we have $\underline{y} = X\beta + \underline{\varepsilon}$, where $V(\underline{\varepsilon}) = \Gamma$ and of course $E(\underline{\varepsilon}) = 0$. We now also assume (for the first time) normality of the residuals. The negative log likelihood function is (except for some irrelevant constants)

$$(20):\ L = \ln|\Gamma| + (y - X\beta)^T\Gamma^{-1}(y - X\beta)$$

Full maximum likelihood estimates minimize this loss function over the unknown parameters.

Let us now specialize (20) to the random coefficient model (4), in the important special case in which all β_{ij} are equal to β_j, and all Ω_{jk} are equal to Ω. Then

$$(21):\ L = \ln|X\Omega X^T + \Sigma| + (y - X\beta)^T\{X\Omega X^T + \Sigma\}^{-1}(y - X\beta)$$

where usually $\Sigma = \sigma^2 I$; however, we use the somewhat more general notation to leave open the possibility of additional parametrization of Σ. The free parameters here are (β, Ω, Σ), although in general there will be restrictions on these free parameters. Some elements of β will be made zero, some elements of Ω will be made zero, and usually we require $\Sigma = \sigma^2 I$.

We can rewrite formula (21) in a much more interesting way, generalizing (and correcting) a result also given by De Leeuw and Kreft (1986).

$$(22a):\ L = \ln\left|\Omega + (X^T\Sigma^{-1}X)^{-1}\right| + \ln|\Sigma| + \ln|X^T\Sigma^{-1}X| + (b - \beta)^T\left\{\Omega + (X^T\Sigma^{-1}X)^{-1}\right\}^{-1}(b - \beta) + ns^2$$

$$(22b):\ b = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y$$

$$(22c):\ ns^2 = y^T\{\Sigma^{-1} - \Sigma^{-1}(X^T\Sigma^{-1}X)^{-1}\Sigma^{-1}\}y$$

Thus b is the least squares (and maximum likelihood) estimate of β in the model $\underline{y} = X\beta + \underline{\varepsilon}$, if $V(\underline{\varepsilon}) = \kappa\Sigma$. Because Σ is usually $\sigma^2 I$, b is usually the unweighed least squares estimate. And ns^2 is the residual sum of squares from this regression (i.e., s^2 is the maximum likelihood estimate of κ). Anyway, it is clear that transforming the likelihood as in (22) means that we can actually work with matrices of order m instead of with matrices of order n, which is almost always a considerable gain in terms of computational complexity. It also shows that estimates of β and Ω are functions of the vector b, the matrix $X^T\Sigma^{-1}X$, and the quantity s^2, no matter what the specifics of the model are. We start by reducing our data structure in this way. The hierarchical model (11) and the multilevel models (16) and (18) can be written in the random coefficient form, with a likelihood function that specializes (21). In all cases full maximum likelihood estimates are computed by minimizing the loss function L.

1.3.2. Restricted Likelihood Function.

It is well-known, from ordinary analysis of variance theory, that ML applied in this way often produces *downward biased* estimates of the variance estimates (Harville, 1977). Variance estimates are generally too small, suggesting more precision than we actually have. In order to understand this properly, remember that the ML estimate of the variance from a sample of a normal distribution is the sum of squares around the mean, divided by n. We know this is too small, because the unbiased estimate divides by n - 1. In ordinary fixed effects ANOVA the ML estimate of the error variance is the residual sum of squares divided by n, which can differ quite substantially from the unbiased estimate that divides by n - p. A solution to this problem is to use n - p linearly independent or orthogonal contrasts H, which are orthogonal to the design matrix, and estimate the residual variance by applying ML to H^Ty and not to y. This produces *restricted maximum likelihood*, or RML, estimates. Another way of formulating this is to compute the negative log likelihood of the ordinary least squares residuals as our loss function. Thus we transform the data to residuals and use the likelihood of the transformed data as our criterion. Let us agree to call our previous estimates, which maximize the likelihood of the observations directly, *full maximum likelihood* or FML estimates.

It has been shown by Harville (1974) that the loss function, which we have to minimize in order to get RML estimates in the general heteroscedastic linear model, is

$$\text{(23a):}\quad L = \ln|\Gamma| + \ln|X^T\Gamma^{-1}X| + (y - Xb_\Gamma)^T\Gamma^{-1}(y - Xb_\Gamma)$$

with

$$\text{(23b):}\quad b_\Gamma = \left(X^T\Gamma^{-1}X\right)^{-1}X^T\Gamma^{-1}y$$

Observe this does not involve β any more. Compare (23) with (20), which can be written as

$$(24):\ L = \ln|\Gamma| + (y - Xb_\Gamma)^T\Gamma^{-1}(y - Xb_\Gamma) + (b_\Gamma - \beta)^T X^T \Gamma^{-1} X(b_\Gamma - \beta)$$

Of course we get additional specifications of (23) in the case of random coefficient models, hierarchical linear models, and multilevel models, but we shall not discuss these specifications here.

1.4 Algorithms

The four programs we discuss use three types of algorithms. One way to proceed is to think of the likelihood function as depending on two sets of parameters: the regression coefficients β, and the variances Γ. If the variances are known, the regression coefficients can be estimated easily by weighted least squares. If the regression coefficients are known, the variance components can perhaps be estimated fairly simply as well. The idea is to alternate these two minimizations iteratively. Oberhofer and Kmenta (1974) already proved the convergence of such an alternating algorithm, but application in a context such as ours was proposed by Goldstein (1986).

If we write down the likelihood equations, using (24), in the way suggested by this algorithm, we find in the first place

$$(25):\ \beta = b_\Gamma = \left(X^T\Gamma^{-1}X\right)^{-1}X^T\Gamma^{-1}y$$

Secondly we can rewrite L as

$$(26a):\ L = \ln|\Gamma| + \mathrm{tr}\,\Gamma^{-1}V$$

with

$$(26b):\ V = (y - Xb_\Gamma)(y - Xb_\Gamma)^T + X(b_\Gamma - \beta)(b_\Gamma - \beta)^T X^T$$

If we replace $(b_\Gamma - \beta)(b_\Gamma - \beta)^T$ by its expected value, then we find

$$(26c):\ \hat{V} = (y - Xb_\Gamma)(y - Xb_\Gamma)^T + X\left(X^T\Gamma^{-1}X\right)^{-1}X^T$$

The alternating maximum likelihood algorithms, now start with an initial estimate of Γ, for instance, one that is proportional to I. Next, (25) is applied to find the corresponding estimate of β; (26a) is maximized over all Γ of the appropriate form, with V given by (26c), using the current estimates of Γ and β. This gives a new Γ. In this last minimization we use the fact that in random coefficient models, Γ is a linear combination with unknown coefficients (the variance and covariance components) of known matrices.

Goldstein (1989) points out that the algorithm can be applied equally well to RML estimation. This result is based on the differential identity

$$(27):\ \partial \ln \left|X^T\Gamma^{-1}X\right| = \mathrm{tr}\left(\left(\partial\Gamma^{-1}\right)X\left(X^T\Gamma^{-1}X\right)^{-1}X^T\right)$$

The advantage of the alternating or weighted least squares algorithms is that the problem is decomposed into a sequence of linear regressions which can be solved quickly and precisely. A major disadvantage is that in each iteration we use the design matrix X, and reductions of the problem by using sufficient statistics (cf. below) are not possible. Another consequence of this is, however, that Goldstein's algorithms can fit a more general class of models such as hierarchical models with nontrivial disturbances on the first level. From the mathematical point of view there are a number of questions that still need to be cleared up. It seems that a precise convergence proof is missing, and it seems to us that one based on majorization (as for the EM algorithm) must be possible. We are also a bit slow to understand the precise implications of (27), but we continue to study them diligently.

The second type of algorithm, proposed for hierarchical linear models by Longford (1987) and for multilevel models by De Leeuw and Kreft (1986), is simply the classical method of scoring. This is the Newton-Raphson method, applied to the likelihood function with a convenient first-derivative approximation to the second derivatives. Full implementations of the Newton-Raphson method for general mixed linear models were discussed by Jennrich and Schluchter (1986) and by Lindstrom and Bates (1988). For random coefficient models we use the transformation of the likelihood given in (22). It shows that we can reduce the problem to one that only involves the sufficient statistics b and σ^2, which means that we can actually throw away the raw data after we have computed these statistics. And this makes it possible, of course, to deal with a virtually unlimited number of individuals. Longford (1987) generalized this particular type of partitioning to more than two levels, using results of LaMotte (1972).

The third type of algorithm is the EM algorithm of Dempster, Laird and Rubin (1977). It was applied to covariance component models by Dempster, Rubin, and Tsutakawa (1981), and to (longitudinal) multilevel linear models by Stram, Laird and Ware (1986). (Compare also Jennrich & Schluchter, 1986; and Lindstrom & Bates, 1988.) Multilevel programs discussed by us that use the EM algorithm are HLM by Bryk, Raudenbusch, Seltzer, and Congdon (1988), and GENMOD by Mason, Wong and Entwistle (1983). The EM algorithm is based on a clever bonding of the likelihood function with a more convenient minorization. In each iteration the minorization is maximized, and each of these steps increases the likelihood function as well.

Let us illustrate this with the case of the mixed linear model $\underline{y} = X\beta + Z\underline{\delta} + \underline{\varepsilon}$. Here $\underline{\varepsilon}$ is $N(0, \Sigma)$ and $\underline{\delta}$ is $N(0, \Omega)$, and the two are independent. The conditional distribution of $\underline{y}$ given $\underline{\delta} = \delta$ is $N(X\beta + Z\delta, \Sigma)$, and thus the log-likelihood can be written in the form (except for some constants)

$$(28):\ L = -\frac{1}{2}\ln|\Sigma| - \frac{1}{2}\ln|\Omega| + \ln\int_{-\infty}^{+\infty} \exp\left\{-\frac{1}{2}\left((y - X\beta - Z\delta)^T\Sigma^{-1}(y - X\beta - Z\delta) + \delta^T\Omega^{-1}\delta\right)\right\}d\delta$$

In order to apply majorization we use the general result that

$$(29):\ \ln\left\{\frac{\int f(u,v)dv}{\int f(t,v)dv}\right\} = \ln\left\{\frac{\int f(t,v)\frac{f(u,v)}{f(t,v)}dv}{\int f(t,v)dv}\right\} \geq \frac{\int f(t,v)\ln\left\{\frac{f(u,v)}{f(t,v)}\right\}dv}{\int f(t,v)dv}$$

This is true for any function f(u,v) for which the integrals and logarithms are defined. It is a simple consequence of the concavity of the logarithm, which in this context is also known as Jensen's inequality. Moreover we have equality in (29) if and only if $u = v$ Now apply (29) to

(28). We use for f(u,v) the joint density of $\underline{y}$ and $\underline{\delta}$ at parameter values (Σ,Ω,β), and for f(t,v) the joint density at parameter values in a previous iteration, written with tildes above the symbols. We integrate over δ, of course. Using probabilistic notation,

$$(30): L(\Sigma,\Omega,\beta) \geq L(\tilde{\Sigma},\tilde{\Omega},\tilde{\beta}) + E_{(\tilde{\Sigma},\tilde{\Omega},\tilde{\beta})}\left[Q(\Sigma,\Omega,\beta)\right] - E_{(\tilde{\Sigma},\tilde{\Omega},\tilde{\beta})}\left[Q(\tilde{\Sigma},\tilde{\Omega},\tilde{\beta})\right]$$

Here Q is the logarithm of the joint density and the expectation is taken over the conditional distribution of $\underline{\delta}$, given $\underline{y} = y$ (the data). Now the only term on the right that depends on the unknown (Σ,Ω,β) is the middle one. We maximize the right hand side by maximizing this middle term over the parameters, which gives us new estimates $(\Sigma^+,\Omega^+,\beta^+)$. From our inequalities so far, $L(\Sigma^+,\Omega^+,\beta^+)$ is strictly larger than the right hand side of (30), which in its turn is larger than the log likelihood in the previous iteration. Thus we increase likelihood, and by iterating this process we produce a convergent sequence of likelihood values and parameter estimates (if several regularity conditions, which obtain in our case, are true). Maximizing the middle term on the right of (30) is easy, because we only need to compute expectations of linear and quadratic functions of δ, and the conditional distribution of δ given y is a known multivariate normal.

Theoretically, a comparison of the various types of algorithms should be based on the fact that EM typically has (very) slow linear convergence, which is global (i.e., which occurs from any starting point). Both weighted iterative least squares and scoring have fast linear convergence, and for models which fit very well, scoring will be almost quadratic. We do not yet understand the behavior of Goldstein's algorithms very well, because the descriptions are somewhat short and we do not have the code. This means that we cannot be too sure about the convergence properties either. The Newton-Raphson method is truly quadratic, but it may diverge from starting points that are not appropriately chosen. In this paper we compare EM (used in GENMOD and HLM), scoring (used in VARCL), and weighted iterative least squares (used in ML2). From our results so far it seems that in practice the results are less clear than the theory above suggests. But precise results will also depend on the details of the parametrization and on the particular expression for the likelihood function that is used.

1.5 Limitations

The multilevel techniques discussed below have various limitations, which perhaps restrict their applicabi'ity to educational research data. First, they are design-oriented, assuming fixed regressors on the individual level. In most evaluation studies, however, the regressors are sampled in the same way as the dependent variables, and thus this assumption is not appropriate. The second limitation is that the techniques deal with a single dependent variable predicted by a number of independent variables. In evaluation studies, and in school research in general, there is often more than one criterion to predict. One could apply the technique to each criterion separately, but this is not satisfactory because it ignores the relations among criteria. A multivariate extension is needed, and, more generally, we would like to have the capability of fitting path models with multilevel techniques. A third limitation is that the algorithms, or at least some of them, cannot handle missing data and categorical dependent variables. A fourth limitation is the linearity of the existing techniques; a fifth is the assumption of normality of the residuals. Although these last two are indeed serious restrictions of generality, developing a technique that does not impose these restrictions involves a more far-reaching generalization of the existing methods than developing multilevel path analysis techniques. Farther on in this report we shall indicate how we want to remove some of these restrictions of generality in the program MULTIPATH. This program will fit path models to data measured at two or more levels of a hierarchy. It will be able to handle categorical endogeneous variables, by treating such categorical variables as discreticized indicators of latent continuous variables (in this respect it is similar to the work of Muthen, 1989).

2. DESCRIPTIONS OF THE FOUR SOFTWARE PACKAGES.

2.0 Introduction

The four software packages are compared in the following way

A	Design philosophy.
B	Implementation details (language, OS, hardware).
C	Models.
D	Routines (algorithms, centering, boundaries, singularities).
E	Data setup and data handling (input, preprocessing, missing data)
F	Results (output files, written output, special statistics).
G	User friendliness (ease of use, manual, examples).
H	Special features (program limitations, unique features, special options).
I	Results and speed.

Direct quotations from the program output will be written as

```
>THIS IS A DIRECT QUOTATION
```

The comparisons mentioned in the first paragraph are collected in a separate chapter.

2.1. GENMOD

A. Design Philosophy

GENMOD is written by Benjamin Hermalin and Albert F. Anderson at the Population Studies Center, University of Michigan, from instructions provided by George Y. Wong and William M. Mason. The program implements the general model proposed by Wong and Mason (1989) (see also Mason, Wong & Entwistle, 1984). It is a (two-level) multilevel model, according to our definition, because it allows the user to specify the macro variables to be used as regressors in a macro regression in which the regressand is the micro coefficient or micro intercept.

GENMOD is developed to accommodate two broad classes of applications: comparative analysis and contextual analysis. Contextual analysis is found in the other three programs. Special to GENMOD is comparative analysis. The assumption here is that we have a different data file for each context; these files may even have different formats. Moreover, the micro data file for one context may also contain variables that are different. This characteristic of the program is very valuable in demographics, the field for which this program was developed. As shown in the original paper of Mason, Wong and Entwistle (1984), countries may differ in their methods of birth control, in the way birth control clinics operate, and/or the extent that they are available in different countries. At the same time, the background variables of the women involved in birth-control efforts may be defined differently in different countries. The program was originally designed to analyze this kind of data, but a later version, dated April 1989, also provides the opportunity to use a single micro file as input and a single associated format statement (as do the other three programs). This kind of data file is used for analyzing data that are similar over all groups.

The program is quite inexpensive ($20.00 at this writing). It comes on two 5.25 floppies and includes source code, documentation, and (hypothetical) examples.

B. Implementation details

The program is written in Fortran 77 and is currently compiled to run under the MS-DOS and MTS operating systems. File names must satisfy MS-DOS file naming conventions. Under MS-DOS, the program reads and writes ASCII files only; MTS uses EBCDIC. The manual assumes that the program is running under MS-DOS; MTS tailoring is given in the Appendix.

There are three versions (GEN30, GEN40 and GEN50), which differ in the size of the real array storage that has been allocated (35,000, 45,000 and 55,000 elements of REAL*8 storage, respectively). The distribution includes source code, however, which means that (at least in principle) any DOS or OS/2 user with a (MicroSoft) FORTRAN compiler can make his own version with storage requirements adapted to his own environment. In practice this may still lead to some problems. We have tried to compile GENMOD with the Leahy FORTRAN compiler under DOS, with the VS-FORTRAN compiler under MVS, and with the Language Systems FORTRAN compiler under the MacOS, and none of these attempts have been successful so far.

The memory available for array storage is the major constraint on the size and complexity of the model that can be handled. The array space required for a given setup is allocated dynamically from a single large array that has been defined in the source code. The three versions of the program differ only in the size of that array. For GEN30 the load size is 420K, for GEN40 it is 500K, and for GEN50 it is 580K. The amount of array space required depends largely on an interaction between the number of contexts to be handled and the complexity of the model. The memory requirements are greatest for the combination of a large number of contexts with a model that specifies fixed effect coefficients and OLS estimation for the start values.

It is possible, in principle, to determine the maximum size of the problem that can be handled from the size of the large array containing all the reals. The manual does not give general rules. Clearly there is no maximum on the number of cases.

C. Models

The basic model fitted in GENMOD is the two-level model of section 1.2.3, rewritten in matrix notation. The first level equation is

$$\text{(GENMOD1): } \underline{y}_j = U\alpha_j + X_j\underline{b}_j + \underline{\varepsilon}_j$$

and the second level equation is

$$\text{(GENMOD2): } \underline{b}_j = Z_j\gamma + \underline{\delta}_j$$

For the disturbances we assume

$$\text{(GENMOD3): } \underline{\varepsilon}_j \text{ ' } N(0, \sigma_j^2 I), \underline{\delta}_j \text{ ' } N(0, \Omega)$$

The matrix Z_j in (GENMOD3) has a block structure, with the macro regressors for each of the G contexts as row vectors in the diagonal blocks. Thus

$$\text{(GENMOD4): } Z_j = \boxed{\begin{array}{ccc} z_{j1}^T & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & z_{jG}^T \end{array}}$$

The notation here differs a bit from the more general notation in 1.2.3 because we can simplify it for two-level models. The notation also differs from the notation used in the GENMOD manual and in the Wong and Mason (1989) paper.

Observe that (GENMOD3) tells us that different contexts may have different first-level error variances. This is, of course, particulary important in comparative analyses. In the same way, from (GENMOD1), we can have different fixed first-level regressions in each context. Both conditions taken together imply that in comparative analysis the contexts are much more loosely coupled by the model than they are in contextual analysis. GENMOD contains the option to require equality of the error variances in groups of contexts, which means that we actually have a simple third level model for the error variances.

D. Routines.

The basic algorithm of GENMOD performs the following three steps:

1. Constructs cross-product matrices and G-matrices from raw data.
2. Performs ordinary least squares (OLS) estimatior to obtain initial values.
3. Performs EM iterations.

The user can activate the three steps by choice, so any of the three can be the starting point of the analysis. To activate step one the user must supply raw data files. To activate step two the user must supply a file with cross-product matrices. To activate step three the user must supply a file containing starting values for the micro and macro error variances and covariance. As a result of the above options the program can provide the user with two output files. One contains the statistical output, the other contains the information that can be used in subsequent analyses to activate the second or third step.

The maximum number of iterations is specified in the batch job. If convergence has not been achieved by the NUMIT-th iteration, the program will nonetheless stop, giving complete output as of the NUMIT-th iteration. The estimation procedure is restricted maximum likelihood (RML). The EM algorithm used is based on equations developed by Mason and Wong.

E. Data Setup and Data Handling.

The program runs as a batch job. To run the program the user must create a Setup File, a Micro File, a Macro File and (optionally) an Auxiliary Data File.

The setup file in GENMOD describes the model to be estimated and provides instructions for reading and saving information. The structure of the setup file is like the usual SPSS or SAS job, with the header being a specified output file name and three options of data input to choose from (raw data, cross products, or macro error variances and covariances). In principle the data setup consists of 14 lines. When the first setup is ready for a particular data set, only little changes are needed in the setup for fitting the following models. The first lines of such a setup, with a data set named GALO, is shown in Figure 1.

```
#1    GALO.OUT
#2    GALO.DATA
#3    **********************************************************
      * This is the first of the two models we want to fit.  We
      * start with a random intercept only, and the next one will
      * also have a * random slope for IQ.  The following variables
      * are all student level variables: IQ, SES and sex.  No
      * second level variables.

#4.1 0    0 2  0    1   1   1
#4.2 GALO.DAT
#4.3 (6F5.)
#5    36  3   3   100
```

The first line contains the name of the output file, followed by a header for the run, followed by as many * lines for the documentation of the job as needed, followed by a blank line. Line 4.1 contains seven numbers in free format, with at least one space between each entry. In this line the following choices are made: OLS as starting values, a report on all iterations and the request to include in the output file the OLS parameter estimates. In the same line we have informed the program that the data is in one single file (instead of separate files for each context) and we request estimation of a pooled variance over all J contexts (instead of the other option, to calculate an unique variance for each context). Line 4.2 contains the name of the file (in which all the micro data are stacked, context by context). Line 4.3 contains the format statement (which may be different for each context, but is equal over all contexts in this example). Line 5 contains, in order, the number of contexts, the number of micro regressors, the number of fixed-effect micro variables, and the number of iterations. Lines 6 thru 13 (not shown here) contain the macro level variables, the macro file name and format, the model to be fitted, and so on, in the same straightforward fashion.

The user starts the run stage by entering the name of the version to run (GEN40, for instance) and responding to the program's request for a setup filename. An additional request may ask for confirmation that the already existing output or savefile be overwritten. The options for data input are (a) to start with raw micro and macro files, (b) supply a file containing cross-products matrices and descriptive information, or (c) activate the program with macro error variances and covariances. The latter data can be provided by an earlier run of the program by using an AUXILIARY file created in the preliminary run as a SAVEFILE. OLS estimations or starting values read from an AUXFILE not constructed by the program also may be used as starting values for micro error variances and for the matrix of macro error variances and covariances. The option to use an AUXFILE becomes very useful when (a) the previous job did not iterate to converge and a restart beginning with the last iteration is desired, (b) the user wants to do a subsequent analysis in which only a subset of the original values is used, or (c) if we want to respecify the model by making a change in the macro equations. In several options it is possible to use the SAVEFILE of a previous run as AUXFILE in a following run. Because LS estimates may be chosen as starting values, they also are presented in the output. If not, no LS estimates are provided by the program.

To analyze the data with GENMOD, two data files have to be prepared: a micro file and a macro file. The macro file should consist of the same number of contexts and the same names (a separate name for each context in case separate files for each context are used) as the records in the micro file. This also shows the uniqueness of the GENMOD computer program, which allows each context to have its own raw micro data file, with variables that may be in different order and even to a certain extent be different measurements. In that case, if there are J contexts, the user must prepare J micro data files before running the program.

In this report we only used GENMOD to analyze one single micro file, that is, the micro data for each context were readable by the same format statement. In such a case the single data file contains all information over contexts, but observations have to be sorted by context so that they are contiguous within contexts. Agreement must exist between the order of the contexts within this single micro file and the order in the other file, the macro data file.

Missing data cannot be handled by GENMOD and, as a result, are excluded. This is true for the other three programs as well.

F. Results

The program can produce two different output files. One of these files contains the statistical output, that is, all relevant parameter estimates calculated by the program. In addition to this file, the program may produce an output file called SAVEFILE or AUXFILE (an auxiliary rile), which contains all the information necessary for restart or for subsequent analysis. If this file is to be used in the next run, it must be saved under a different name so that it is not overwritten by a file of the same name in the next run. (In GENMOD the SAVEFILE is an ASCII file, thus it can be edited easily before it is used.)

The program writes the results of its operations into a formatted output file with numbered pages. The output file is described in Appendix F and consists of the following information:

1. The setup parameters.

2. OLS estimates of the micro parameters and cross products are printed on request if starting values are estimated during the run in case raw data are the input data. (Note that if cross products, variance-covariance matrices, or other starting values are entered, no starting values have to be estimated. As a result no starting values are printed in the output file.) KC is the number of fixed-effect micro variables included in the model. If KC = 0 the estimated variance-covariance matrices of the micro parameters are printed.

3. OLS estimates of eta, tau, sigma-squared per context, and alpha per context.

4. Information about convergence at every iteration.

5. The RML estimates of sigma-squared and omega at the last iteration, as well as the estimates of gam , the slopes per context, and the intercepts per context. The output provides posterior means for each context, with the standard errors.

In sum, the output consists of the usual basic information: information about convergence at every iteration, and RML estimates of the sigma-squares and omega parameters at the last iteration, as well as estimates of all the parameters. More can be obtained on request.

The estimates of the coefficients (gamma) of the micro-macro inten tions are provided with the usual standard errors (asymptotic standard normal variates under the model), the covariance matrix, and the standardized covariance matrix (equals the correlation matrix). The slopes and intercepts for each group are reported with the usual standard errors and the estimated variance-covariance matrix. Two different estimates for the regression coefficients per context are given: the OLS estimates and the ML posterior means. Not provided are t-test values for the parameters, or chi-square tests.

G. User friendliness.

The software is in some respects easy to use and unproblematic, but in others somewhat puzzling. The present version of the manual is not very clear in how and when to use certain options (the authors are working on this). A user well introduced in the problems and possibilities of multilevel modeling may encounter fewer problems than will a novice. There is no example of the output files and no explanation of the outcome in the manual (although there are examples on the disks). For instance, LS estimates of the regression coefficients (called alpha and beta) are given for each context, but it is unclear from the manual and/or output what alpha and beta represent. Some comparable problems can be found in the (R)ML output. Is the posterior mean given in the alpha vector for a group the macro level disturbance (two random coefficients produce two macro level disturbances, three produces three, etc.)? Are the posterior coefficients (beta) of the posterior variable-effect micro variables equal to the means for the intercept and the slope(s)? Probably yes, but the manual does not tell us. And, for the values between brackets, are the t- or z-values posterior means or the raw standard errors? Another example: It is hard to find how to declare the dependent variable in a batch job. Even worse, we deduced the position of the dependent variable from an example and still do not know where to look in the manual for this information. Together with the fact that no examples are provided and no output file is presented in the manual, GENMOD is not easy to use for the first time. The input in a batch job is otherwise fairly straightforward, but the preparation of the raw data file for input needs some thought. Mainly because the program is set up to analyze different types of contexts the input file needs a special structure.

Definitely user-friendly are the error messages which give you a clear idea what goes wrong. The program attempts to trap all setup and data errors that could cause the program to fail. The error messages are intended to help the user pinpoint where and why an error occurs, but the nature of the program, the number of data files that must be handled, and the nature of the EM algorithm can combine to produce very perplexing consequences from fairly simple setup errors.

A complete list of all error messages (number 1 to number 52) is added to the manual. Errors are generally reported on the screen and in the output file, but the screen output gives a better trace of the program's progress in reading and interpreting the setup file. Examples:

```
ERROR 23: The number of contexts specified in the Auxiliary
Data File (11) and the Setup File (5) do not agree.

ERROR 34: The macro data file, Macro-File-Name '....'., could
not be assigned. Check the filename and status.
```

The manual advises the user to rerun a setup that produces errors (with the screen output redirected to a file) and verify if the program is interpreting the setup file as intended.

The program comes with one artificial data set that consists of 10 groups. There are 5 groups with 10 observations and 5 groups with 20, and there are 5 micro-variables. The disks contain 10 setup files for 10 runs with these data, complete with output, and explanations in the headings. This is actually quite useful to show the various options of the program, although it gives no clear indication of possible applications.

H. Special Features

Because the program is running as a batch job, only one model per run can be fitted. The testing of fit among several models is not possible by directly comparing deviances, because the latter are not reported. But it is possible to deduce the difference in fit among models from the reported log likelihood (the deviance is -2 times the log of the likelihood). The log likelihood of Restricted Maximum Likelihood (RML) and the Full Maximum Likelihood (FML) are both reported in t... output file of the software.

GENMOD was developed for *comparative analysis*. In this type of analysis different files for each context can be used, and these files may even have different formats. Moreover, the micro data file for different contexts may also contain different variables. This makes it possible to fit *different estimates of the variance per group* (the sigma-squared). In the other three software programs discussed in this report, this variance is assumed to be equal over groups and is not allowed to vary. In the case of the comparative analysis, the output of GENMOD produces regression coefficient estimates and sigma squareds for each context separately, as well as OLS estimates and the posterior variable-effect micro coefficients with the respective standard errors.

2.2. HLM

A. Design Philosophy

HLM, Version 2.1, has been written by Anthony Bryk, Steven Raudenbush, and Richard Congdon. The manual was written by the same authors, with Michael Seltzer (1988). It is the most popular program in the USA for at least two reasons: the easy-to-use interactive interface, and the output which includes significance tests, model testing, and other desirable properties Another explanation is the educational character of the manual. It provides a theoretical background for multilevel modeling and an abundance of references for more study. The introduction explains why :ınd how a hierarchical linear model is useful in many research situations.

HLM is developed to accommodate two broad classes of applications: contextual analysis and growth curve analysis. One application of the contextual analysis is for research on school effects, where the first level represents within-class or school analysis on student data, and the second level represents the between-class or school analysis. An application of growth curve analysis is for repeated measurement data. The first level in the latter case is the individual change, and correlates of change, in a within-person model. The second-level analysis is the estimation of the effect of environmental characteristics upon these individual growth curves (see Raudenbush, 1989; and Bryk & Raudenbush, 1987). This last type of analysis allows the researcher to: (a) describe the structure of the mean growth, (b) estimate the extent and character of the individual growth in comparison with the mean growth, (c) assess the reliability of the study of these growth curves, (d) estimate the correlation between the measures over the same person, in the same way that the correlation between students in the same school is measured, (e) assess the reduction in the unexplained parameter variance, and (f) improve predictions of future individual growth. We know that hierarchical linear models can handle repeated measurement data, but since specialized programs for this type of problem have been available for a while and in more advanced stages of development, we see no specific reason to prefer HLM models for this kind of data. A later report (Van der Leeden & de Leeuw, in preparation) compares HLM with two software packages that were written with longitudinal data in mind: BMDP-5V (Schluchter, 1987) and GGCMAOV

(Stram, Laird & Ware 1986). In the present report we compare HLM in the usual way, with hierarchical linear data and no repeated measurements.

B. Implementation details.

There are basically two versions of the program available. The first one is for workstations or mainframe computers with no real restrictions on the memory. The second one is an adaptation for the PC, and it takes the 640 K memory limit of DOS into account. Both versions are written in FORTRAN 77, although for the DOS version the main program and some screen control functions have been written in C. This mixed language feature of HLM, plus the particular type of screen control used, make the program less than completely portable, although the computational routines are in straightforward FORTRAN 77. The source of HLM 2.1 is no longer in the public domain, which means that the question of portability is no longer very interesting. We know of a version for UNIX on HP workstations, a version for DOS and OS/2, we know that older versions of HLM have been compiled under VS-FORTRAN to run on IBM mainframes under CMS or MVS/TSO. We have worked quite hard on a Macintosh version of HLM 2.0, and although the program compiled (using Language Systems FORTRAN under MPW), we could not get it to run.

Current program limitations for the PC version (manual, p. 39) are as follows: There is a maximum of 10 within-unit variables per model. The input file and sufficient statistics file can contain 25 within-unit variables, 25 between-unit variables, and 300 units. In the between-unit model there is a maximum of 15 variables per equation, and the maximum on the total number of fixed effects over all equations is 35. For non-PC versions the default limitations are a bit higher, and if the parameters statements in the source code are modified, all limitations can, in principle, be removed.

C. Model

The basic model fitted in HLM is again the two-level model discussed in section 1.2.3. We rewrite it in the same way as for GENMOD:

$$\text{(HLM1):}\ \underline{y}_j = X_j\underline{b}_j + \underline{\varepsilon}_j$$

and the second level equation is

$$\text{(HLM2):}\ \underline{b}_j = Z_j\gamma + \underline{\delta}_j$$

with the structure of Z_j the same as in GENMOD. For the disturbances we assume

$$\text{(HLM3):}\ \underline{\varepsilon}_j\text{'}\ N(0, \sigma^2 I), \underline{\delta}_j\text{'}\ N(0, \Omega)$$

Thus the micro level errors are the same for all contexts. The notation differs again from the notation used in the HLM manual, because we prefer to choose a uniform notation to desciibe all programs.

By default, both the micro model and the macro model have an intercept, but the default can be overwritten. For growth curves, for example, it can be interesting to fit models without a micro intercept. At the time of the run the user can introduce additional restrictions on the parameters. Some gammas and some omegas can be set equal to zero (only the diagonal elements can be set for the omegas; this implies that all off-diagonal elements are zero). Thus we can have micro variables with only a fixed effect and micro variables with only a random effect, the default being that a micro variable has both types of effects.

D. Routines.

Two routines are given in the manual: (a) an EM algorithm with an Aitkin accelerator, used as the core routine in the HLM program, and (b) a V-known routine. The latter is a more specialized algorithm, which assumes that there is a fixed number of random parameters for each context whose dispersion matrix in each context is known. The V-known routine is useful mainly in research synthesis (*meta analysis*), and we have not incorporated it in our comparisons. See Raudenbush and Bryk (1985) for details. There is also an application of the V-known option in the HLM manual, section 2.8.

In the more general contextual analysis, the dispersions are unknown, and they are estimated jointly with the other parameters. To generate starting values for sigma-squared and omega, HLM uses an ANOVA-type procedure. When the starting procedure fails to produce a positive definite omega matrix, the automatic fix-up routine, used in earlier versions, is replaced in the recent version (HLM 2.1) by the following options: (a) the decision to stop the analysis and to quit, (b) setting all the diagonal elements to zero, (c) resetting these values by plugging in the users' own solution, or (d) trying the original solution with an automatic fixup. The differences among the above solutions (except the first one, "to quit") is mainly in the number of iterations needed for subsequent estimates of omega and sigma-squared.

The number of iterations is, as usual, optional and left to the decision of the user. The suggested number of iterations for exploratory analysis is 10. In most packages the advice is similar: up to 10 to 15 iterations. In the third part of the report we will show that for EM algorithms this number is definitely too small to get final results that are comparable to those of other programs.

E. Data Setup and Data Handling

Two input files are needed: a within-unit file and a between-unit file, each with an identification number (ID). The within-unit file should contain all the unit level records that are used (in the first analysis) or will be used (in subsequent analyses). The between unit file should contains the group-level records. The raw data input file can be either a "V-known" file or a SYSTAT.SYS file. In case of a SYSTAT input file, the residual file (which is produced by the program) will also be a SYSTAT file. A V-known file is one single unit file with parameter estimates for each context and their associated sampling variance/covariance estimates.

As in the other programs, missing data are not allowed and have to be dealt with before starting the analysis. The program offers (only for the within-unit file) the two usual options: pairs or listwise deletion. No missing data in the between-unit file can be handled within the program. The input file holds all the information of first and second stage units together. The start of the program has a build-in check for identifying missing data and inconsistencies in the data. For example: a group ID is identified, student data exists, but no school data are found. The program gives a warning of this nature. Another inconsistency could be that the ID of a student is available but no student data follow, while group data are available for this student. Again, the program has warnings for this case.

Special options.

There is a choice between two different estimation procedures: "H" for a complete hierarchical model or "M" for a mixed model. In the hierarchical model (H), all within-unit parameters are assumed to be random; in the mixed model (M), one or more of the within-unit coefficients can be treated as fixed. Because of the disadvantages of the hierarchical model (see p. 11 of manual), the mixed model estimation will be the preferred choice for most applications. When choosing this mixed model, one of the questions asked by the program is:

```
>DO YOU WISH TO SET ANY OF THE RESIDUAL PARAMETER VARIANCES
TO ZERO ?
```

If the answer is "Yes" the program asks the user to specify the variables. All variables withi residual parameter variances set to zero are treated as fixed parameters.

In the new versions of HLM (starting with 2.0) it is possible to introduce in the model a variable that has no fixed part, only a random one.

It is possible to center or not center variables from their respective group means within the program itself. Centering from the overall mean is not an option and has to be managed before starting HLM. To center none, all, or one or more variables is a choice left to the user. The option to center or not center the variables is different for each of the several computer programs discussed in this report. Centering around the group mean is, in principle, fitting another model (the so-called *Cronbach model*) and will not be used in the examples in this report. Compare the discussion in sections 1.2.4 and 2.2.

The within-unit models can be estimated with and without intercepts. In other words, the models can produce either standardized or unstandardized coefficients. The relevant option is *suppression of the base*, which means that no intercept is available to interact with the second-level variables.

It is possible to give different weights to the between-unit variables. This is useful in cases when certain types of units (for instance special schools) are overrepresented in the sample.

A special default file, DEFILE.HLM, which can be switched on or off, is automatically used if the user does not specify a default file on the command line. This file turns off all special options, thus making a straightforward analysis much easier to handle. When users wish to test (but not fully explore) a specific model, they do not have to respond to irrelevant prompts while using the interactive mode. If the user includes another file instead of the default file, the DEFILE.HLM is ignored and options are available.

Many possibilities for exploration of the data are available. This exploration can be done in three stages: *before*, *in the middle*, and *after*.

Before. In the initial phase, the program can be used to explore all kinds of fixed effects by examining the means, regression coefficients, and ANOVA estimates. Measures of reliability and homogeneity are printed, giving information about the within-unit variables and whether it may be rewarding to include them as random effects in the within-unit analysis.

In the middle. Another exploration is possible for the between-unit variables. The option given here (see the manual) allows the user to examine promising candidates for inclusion in the between-unit model. The output can produce three different matrices: a correlation matrix among univariate regression coefficients (with standard deviations), a matrix of correlations among group-level variables, and a correlation matrix between group-level variables and the univariate regression coefficients.

After. The last exploration of the data can be performed after the first analysis is finished. This is the (optional) hypothesis testing phase (see pages 65-66 of the manual). The exploration includes:

1. A homogeneity test for residual dispersions.

2. A test against an alternative model of the variance/covariance components. In fact, this is the difference in deviance between two models with the same fixed effects in relation to their difference in degrees of freedom and/or the number of parameters. This difference follows a chi-square distribution.

3. A multivariate hypothesis testing for the fixed effect. The residual file is part of the optional output (see section F, "output files," for more information) and can also be further explored by packages such as SAS or SPSS/X to check the adequacy of the fitted model and the assumptions.

F. Results

The output of the gamma coefficients is similar to the output of the other software packages. It provides the user with the estimated coefficients and their standard error. The t-statistic and p-value of each coefficient is a feature not provided by the other packages. These statistics have the same interpretation as in the OLS output for regression coefficients and their reliability.

Three output features are unique to HLM: A reliability estimate, a deviance statistic, and a chi-square statistic.

Reliability estimates for the variables in the model are calculated as a proportion. This is the proportion of the total variance in the within-unit OLS estimates that is parameter variance, similar to the ratio of true and observed variance. For completeness we give some of the formulas that are relevant. The variance of the OLS regression coefficients is

$$\text{(HLM4): } \mathrm{VAR}(\hat{b}_j) = \sigma^2(X_j^T X_j)^{-1} + \Omega = V_j + \Omega$$

The reliabilities are on the diagonal of the matrices

$$\text{(HLM5):}\quad \Lambda_j = \Omega(V_j + \Omega)^{-1}$$

Using the matrix Λ_j we can write the posterior means of the regression coefficients in the form

$$\text{(HLM6):}\ b_j^* = \Lambda_j\hat{b}_j + (I - \Lambda_j)Z_j\gamma^*$$

where

$$\text{(HLM7):}\quad \gamma^* = \left(\sum_{j=1}^{G} Z_j^T\left(\sigma^2\left(X_j^T X_j\right) + \Omega\right)^{-1} Z_j\right)^{-1} \sum_{j=1}^{G} Z_j^T\left(\sigma^2\left(X_j^T X_j\right) + \Omega\right)^{-1}\hat{b}_j$$

One important reason for the success of the two-level model, and of HLM in particular, is the basic simplicity and interpretability of these formulas.

The *deviance* statistic tests the fit of the model. The deviance compares the full model (with all the first-level parameters random and all the second-level variables as functions of the first level parameters) with a more restricted model of some sort. The deviance is reported with the number of the degrees of freedom. Comparing the two models will determine how much fit the researcher will lose by releasing some parameters (by setting the variance equal to zero or leaving the variable entirely out of the analysis).

The *chi-square* statistic tests if a sufficient part of the variance of a particular coefficient is explained by group characteristics or if a significant part is still left unexplained. This statistic can be used to examine research hypotheses. The chi-square test is an approximate test and does not take the full multivariate structure of the estimation into account. The deviance test is, for that matter, a more powerful test of the same kind.

Depending on the option chosen by the user, either all or only the first 10 OLS-estimates for each unit are presented. The same holds true for the number of iterations: either all or only the first and the last iterations are printed.

An option is available for setting up a residual output file for later data exploration. The residuals written to this file are: the empirical Bayes (EB) residuals, the OLS residuals, and the fitted values for each within-unit component. Also Mahalanobis distances within each unit are written to this file. The Mahalanobis distance is a measure of how far the observed values of the responses for a given subject are from their estimated conditional means. Assuming multivariate normality, in large samples the Mahalanobis distance will be distributed approximately as a chi-square variate with degrees of freedom equal to the number of responses present for that subject. Large positive values for standardized residuals help detect possible observations that may be outliers.

The output also contains the variance-covariance matrix of the within-unit slope estimates, plus the correlation matrix between these coefficients. This is a convenient way of getting an impression of how much multi-collinearity (if any) exists among the variables of the first level. The slope/intercept correlation is usually high, which leads some researchers to prefer to center variables around the group mean before starting the analysis.

An exploratory analysis can be selected by the option `ADDITIONAL PROGRAM SPECIFICATIONS`. Residuals of the fitted model are used for regression on between-unit variables selected for subsequent inclusion in following HLM runs. This is useful to identify variables that show some relation with other variables not yet explored, and that may subsequently be entered into the next equations.

G. User friendliness.

For the convenience of the user the program provides three data sets: (1) the rat data, (2) the High School and Beyond data with a code book, and (3) a meta-analysis data set of research on teacher-expectancy effects.

The program is completely interactive, which makes it very easy to use. The manual is also user friendly. Every prompt of the interactive session is explained in the manual. It also contains the output of several runs with different data sets; these examples are extensively annotated.

The manual contains useful suggestions for several methods of data exploration. The option to specify an output file for residuals is one of them; others include the directions on how to create a SYSTAT-file or a SAS-file from this residual output file, and the explanation and directions on how to use this file for analyses such as checking for cutliers, normality, or systematic trends in the residuals.

The organization of the manual, easy as it is for first use, also has its setbacks. Specific information is not easy to find, since it is not organized under special headings. Special remarks and basic information are interwoven with explanations of prompts and with examples of different kinds of output. For example: In section 2.7 the program limitations and suggestions for enhancing program limitations are discussed below the prompt `PLEASE ENTER THE BETWEEN-UNIT VARIABLES YOU WISH TO USE.` Section 2.7 is a work session designed to demonstrate all HLM options—program limitations are hard to find. Another example is the procedure for handling boundaries, which is hidden (and explained) below the prompt `WHAT IS THE NAME OF THE OUTPUT FILE.` It is a bit hard to understand the logic of this organization. The organization is most problematic when the user is quite familiar with the program and does not need to read through the prompts anymore, but still needs the kind of information mentioned in the examples. Something similar happened to us when we were analyzing our examples with the GALO and GRAY data. Below the output of the reliability estimates the message appeared, indicating that fewer groups had been used to calculate these estimates. However, no other package calculates reliability estimates; the manual offers an explanation of why and how this happens on page 50. Since this explanation was hidden in an example of an output file, it escaped our attention and we had to consult one of the authors before we could make use of it. In the same way, the program limitations for the PC version are given in a session log on page 39 of the manual. We think the manual would be much improved by a chapter with clear reference and organization on error messages and other miscellaneous happenings that may worry the user.

A nice feature of HLM is that the output can be limited to the essentials. The default options in DEFILE.HLM can be used to drop all but the most essential parts of the output, which turns off all special options and features (such as data analysis for the purpose of preliminary exploration before starting with the definition of the model for hierarchical linear analysis). If the researcher knows which variables to use, this shortcut presents the user with fewer prompts (when using the program interactively), saving her from responding repeatedly to irrelevant prompts. Still, the output provided by HLM is extensive.

The last point to mention here is the very useful list of keywords and options for default files presented at the end of the manual.

It is clear from the options available and the suggestions given that the program was designed to provide a maximum of answers to possible questions and/or wishes posed by a variety of researchers.

H. Special Features

HLM is the only one of the four programs that delivers a variety of tests. These are: (a) the t-test for significance of the fixed parameters, (b) a chi-square test for residual unexplained variance in the first level parameters, (c) a reliability estimate of the first level variables, (d) the three hypothesis tests mentioned before and (e) a test for homogeneity of variances. Homogeneity of variances is assumed under the recent version, which means that a constant error variance is estimated that is assumed to be equal over all schools.

Three options are available for data input. Two of them are unique to HLM. One is the possibility of using SYSTAT files; input of a SYSTAT file instead of an ASCII file is also possible. For people who have SYSTAT this provides additional possibilities for data handling. Although it is not clear that SYSTAT is a particularly good choice in this context, it certainly is nice to have the additional option. The other unique input option is the *V-known* file, which we discussed briefly in section D of this chapter.

2.3. ML 2

A. Design philosophy.

ML 2 is software for two-level analysis (we expect ML 3 for three-level data soon) by Rabash, Prosser and Goldstein. ML 2 is based on earlier work by Goldstein (1987). It was produced as part of the Multilevel Models Project of the Institute of Education at the University of London. This project is funded by the Economic and Sociai Research Council of the United Kingdom to extend the theory of multilevel modeling, to study the practical application of the models to real data sets, and to disseminate information about the theory and practice of this form of analysis. ML 2 also was developed for fitting mixed linear models. The program is able to fit data with a two-level hierarchical or a nested structure. The coefficient of any explanatory variable may be random. Among the specialized models that can be estimated using the program are growth curve models and hierarchical logit models.

B. Implementation details.

ML 2 is provided only in binary form. It runs on DOS/OS2 computers and needs 540 K of RAM. Program and manual are a commercial product and can be obtained from the authors. The remarkable aspect of the ML 2 implementation is that the multilevel software is merged with the kernel of the general-purpose package NANOSTAT (Healy, 1987), which offers a whole set of data handling and data transformation operations. NANOSTAT also provides descriptive statistics and high-resolution plots. If you want to use the graphics that come with the NANOSTAT package, you need a CGA, EGA, VGA, or Herculus card. We have had only minor problems with ML 2 (in part III we shall report some bugs, which we presume have been corrected in the newest version).

An important difference between ML 2 and the other three programs is that the data are not first reduced to sufficient statistics and then kept in core memory. In ML 2 the complete data matrix is read into core memory, which means that the restrictions on the size of the problem are much more serious for ML 2. The manual gives no clear cut rules, but on page 107, in the footnote, we find: "Insufficient memory is the most common problem new users experience. Use the CHKDSK command to determine the amount of RAM that is free, and if necessary, remove RAM-resident utilities." In more practical terms this means that really big examples cannot be analyzed by ML2.

C. Models.

The basic model fitted in ML2 is again the two-level model discussed in section 1.2.3. We rewrite it in the same way as in GENMOD and HLM, but we introduce a slight generalization. The first-level equation is

$$\text{(ML 2: 1): } \underline{y}_j = X_j\underline{b}_j + H\underline{\varepsilon}_j$$

and the second-level equation is, as usual,

$$\text{(ML 2: 2): } \underline{b}_j = Z_j\gamma + \underline{\delta}_j$$

For the disturbances we assume

$$\text{(ML 2: 3): } \underline{\varepsilon}_j\,'\;\; N(0, \Omega_1), \underline{\delta}_j\,'\;\; N(0, \Omega_2)$$

Of course the $\underline{\varepsilon}_j$ are of level n, while the $\underline{\delta}_j$ are of level G (the number of contexts). Observe that the two-level model in ML 2 allows for a more general error structure than either GENMOD or HLM. Also it is useful to emphasize that the columns of H_j are not necessarily different from those of X_j. In the single equation specification of the model

$$\text{(ML 2: 4): } \underline{y}_j = X_jZ_j\gamma + X_j\underline{\delta}_j + H\underline{\varepsilon}_j$$

some columns of X_j may be the same as those of H_j, leading to possible identification problems. As usual, our notation differs from the notation used in the ML 2 manual.

D. Routines.

An iterative generalized least squares (IGLS) algorithm provides estimates of model parameters and, when normality assumptions are met, maximum likelihood estimates. ML 2 can also compute unbiased or restricted RIGLS estimates, which are called restricted maximum likelihood (RML) estimates in other contexts. The user has the choice between IGLS, described by Goldstein (1986, 1987) and RIGLS, described by Goldstein (1989). IGLS is the default estimation method and is comparable with FML used in the VARCL program by Longford; RIGLS is comparable with RML as used in HLM by Bryk et al. The Restricted IGLS adds bias correction terms on each iteration to IGLS estimates. This amounts to treating the fixed coefficients as quantities which have uncertainty "built in" when computing the random parameters. IGLS, on the other hand, treats the fixed coefficients as known in the same situation. For small data sets the different estimation procedures may produce considerably different results. Compare our discussion in part I and the data analysis using a small data set in part III of this report. It is not clear for the user when to use IGLS and when to use RIGLS. The fact that the distinction between the two is not really discussed is an omission general to all four software packages, but it is especially missed in ML 2, since the choice between the two is stressed.

The first step in ML 2's estimation process is to conduct an OLS regression using all the cases, ignoring grouping. The estimates for the fixed coefficients and for the residual variance become default starting values for the IGLS or RIGLS iterations. A user having some knowledge of estimates from modeling a particular data set may wish to start the iterations for a new model at a point thought to be closer to the new final estimates than the default values. (See the section on special options below.)

The number of iterations can range from 1 to 999. The default value is five iterations. This number is sufficient for reaching convergence when the conditions are favorable, that is, when the number of observations per unit is large enough to obtain stable estimations, the number of parameters to be estimated is small, and the tolerance/convergence criterion is between 2 and 4. It is advisable to increase the number of iterations when the convergence is reached slowly, the amount of data is small, and/or the number of parameters to be estimated is large.

When one of a pair of random coefficients has a variance of zero, the covariance is set to zero. This default option can be switched off. (See the section on special options below.)

E. Data setup.

The steps in the analysis are Input, Model Specification, Run, and Output. The input file can be either raw data or a modified data set. Only one data file is needed with all micro and macro data together, plus the created interactions of micro and macro variables. In this respect ML 2 is comparable to VARCL, where interactions also have to be planned in advance. In ML 2 the interactions can be made in NANOSTAT in the model specification stage, which makes the data handling easier than it is in VARCL. In cases where variables are centered around the grand mean, this may lead to different results between the combination of VARCL and ML 2 and the combination of HLM and GENMOD, since the last two packages make interactions during the run and, as a consequence, these interactions are not centered. The micro data and interactions have to be sorted by context and two IDs are needed, a case ID and a group ID. Missing data can be handled, but have to be assigned a numerical code, which means that they cannot be left as a blank. The NANOSTAT package allows all kinds of data modifications before starting with the multilevel model specification stage. The model specification stage involves the usual things: the roles of the variables that have been read or created and the parameters to be estimated. In the 'run' stage the user has control over the usual features of the program's estimation process prior to starting computation; some default values can be changed, such as the maximum number of iterations allowed before the program stops, the size of the convergence criterion, and the estimation algorithm used.

The choice of the model is completely free. The user can consider models ranging from a random intercept to a full model with all first level variables random. As in HLM, this program allows variables that are not in the list of fixed effects to be in the list of variables with a random effect. In other words, variables can have a random effect at the second level, while not having a fixed effect estimate at the first level.

As in the other programs, no missing data are allowed in the multilevel modeling stage. The usual preprocessing options are available in the NANOSTAT package: use listwise deletion, input estimated values, or mark with missing value codes.

F. Results.

By entering the command LOGO at the start of the analysis, the user can get a complete log of all information that appears on the screen during the analysis.

Since this default output is minimal, special output can be required by using special commands. The default gives estimates of the fixed and random parameters and the covariance matrices for these estimates. OLS estimates and/or predicted values for each case can be obtained by giving the command OLSE. Optional commands, for instance, allow the user to examine residuals, to obtain confidence intervals, and to do hypothesis testing by contrasting certain parameters. In this kind of hypothesis testing more than one command is needed, since contrasts must be specified (see manual, page 72).

ML 2 provides numerical output in two forms: One labeled listing for interpretation, and one unlabeled listing in a worksheet for graphing purposes. Plotting and graphics commands are explained in the manual (see section 4.3 and subsection 5.4.1). The parameter estimates are presented in the output with their associated standard errors. Each random coefficient will have a set of residuals associated with it for each level at which that coefficient is random. So ML 2 provides predicted values, 'raw' level one and level two residuals, and their comparative and diagnostic standard errors.

In sum, the following results can be obtained, but are optional:

1. Estimates of residuals from each level and their standard errors. These estimates can be used for (a) calculating posterior means or shrunken residuals or (b) for checking model assumptions.

2. Standard deviations of the residuals useful for diagnostic plots (an extensive graphics component, built in the general-purpose package NANOSTAT, enables the user to produce

a variety of plots of the estimated quantities).

3. Predicted response variable values for individuals.

4. Estimates of contrasts of the fixed parameters, associated chi-square values for hypothesis tests, and simultaneous confidence intervals.

The RESI command, used after convergence is reached, stores the residuals requested with this command in columns to be specified by the user. A composite residual for each level can be obtained, if desired, by subtracting the column of IGLS or RIGLS predicted values from the column of observed values of the response variable. Another way to use the residuals is to plot a graph of residuals against predicted values. The user's guide explains the theoretical background of residual structures, as well as its practical application. A convenient feature of the output is that the previous estimate is provided next to the current estimate, making the changes easy to see.

The residuals can be used in two ways. First, the level two residuals may be, and often are, used directly to provide estimates of the unknown level two effects. Second, for hypothesis testing, ML 2 can calculate simultaneous confidence intervals for a set of contrasts that the user specifies, as well as a confidence interval for each contrast considered individually. Each random coefficient has a set of associated residuals for each level at which that coefficient is random. Clearly, the total residual is partly a first level and partly a second level residual. When the total residual is defined as the difference between the estimated and the observed dependent variable, it is possible to test complex hypotheses about several elements of the matrix omega. By using a contrast matrix (equivalent to the matrices used in the Scheffé test for effects in ANOVA), several combinations of effects can be tested against one another without enlarging the alpha level

G. User friendliness.

In the last chapter of the manual, Chapter 9, several examples incorporating different data sets and different set-ups and computer output are given. The examples illustrate the variety of models that can be fitted. The first part of the chapter (9.1) presents a sequenced exploration of the data and a random regression model. The second part (9.2) contains an example of a model with longitudinal data, and the third part (9.3) gives a repeated measures analysis. The last example in 9.4 describes an analysis with proportions as the dependent variable.

Several data sets are used in paragraph 9.1. The first one is an educational example: data on students in 96 inner London secondary schools are used to illustrate various aspects of command usage in NANOSTAT and also to show how to specify a variance component model and a random coefficient model. Three stages are distinguished and explained in the modeling phase of this example: the specification of the model, the run stage, and the output stage. The second example (page 85) is a growth curve analysis, comparing weight gains of young Asian and non-Asian children (n=568). Eight exploratory variables are available at the first level. This example shows how to fit a polynomial, how the program handles a missing value code (such as -1) and how to recode a variable to make it a dummy variable. The third example uses four repeated measurements of jaw bone length from 20 boys at ages 8, 8.5, 9, and 9.5 years of age (see Elston & Grizzle, 1962). The third example shows a pooled cross-sectional time series design. The data is taken from a study of youth unemployment (Garner, Main, & Raffe, 1987). The level one units are the four cells of formed by two binary variables, 'sex' and 'qualifications' (n= 401 units). The level two units are the postal code sectors (n=122 units). The dependent variable is the transformed proportion of employed leavers for a cell within a zone. The output examples in Chapter 9 are useful, although they are perhaps a bit specialized. In his papers and his book Goldstein illustrates multilevel modeling by discussing several relatively small models in detail, not a general model in a general notation first, with the specializations later. The same strategy is followed in the ML 2 user's guide.

The final version of the manual is a nice and complete document: informative and clear as far as the theory and the examples go. It is actually more than a manual, since it introduces the reader to the hows and whys of multilevel analysis with a multitude of references. A disadvantage of the manual is the belated instruction on how to do a multilevel analysis. After the introduction to multilevel analysis in a conceptual as well as theoretical part, the reader has to cover fifty pages (until page 63) before the multilevel data analysis is reached. This is due to the extensive documentation of the NANOSTAT package. Since in most model fitting cases the user may need the use of NANOSTAT commands (illustrated in Chapters 3 and 4), we do not know how the authors could have prevented this circuitous route. Apart from that, the manual is well organized and clear. Each chapter starts with an overview of the chapter's content and a summary of the commands recorded in that chapter. A general overview of all the commands (which includes references to the chapters in which they are discussed) is given at the end of the introduction of Chapter 1. The manual's organization does not prevent a feeling of being lost when starting to use this package for the first time. "What do I do to fit a simple random coefficient model without data manipulation?" is not an easy question to answer, since no examples are given.

As we mentioned earlier a disadvantage of the manual, as we experienced it, is the overload of information on data manipulation and the limited amount of clear information about the most essential aspect of the program: the fitting of a simple multilevel model. Only after the user is familiar with NANOSTAT and its command structure can she discover how easy and simple the model specification can be and how helpful the package is for data manipulation. We will illustrate this with the following example. Four different models are fitted with a data set that consists of only one micro variable and one macro variable. The first step is to prepare the data set and to add all the relevant interactions, which is in our case only one: the interaction between the micro and the macro variable.

Example: (comments are added in parentheses).

```
    DREA 1 C1-C4                (data set, see Chapter 3)
    (F2.0,3F7.3)                (the format)
    MULT C3  C4  C5             (make a new variable by multiplying variable at C3
                                with C4 and put it in column C5; see Chapter 4 for
                                information on data manipulation)
    PUT  3691  1  C6
    NAME  C1  "ID"  C2  "Y"     (all variable names are specified including the new
                                interaction, see Chapter 3).
IDEN  1  'ID'                   (identification number level 1, see Chapter 7)
IDEN  2  'ID'                   (identification number level 2)
RESP  'Y'                       (our response variable has the name Y, see
                                Chapter 7)
EXPL  'CONS'   'X'   etc.       (cons is for constant and X is the name for our first
                                level variable, second level variables have to be
                                specified here, too).
MAXI  100                       (the maximum number of iterations is specified here;
                                see Chapter 8 for this and the following commands)
TOLE    3                       (the precision, here up to three decimals)
BATCH                           (meaning no stopping between iterations)
```

All these commands are put in by the user line by line. The answer given by the program is either an error message (when wrong) or a confirmation (when right). The user realizes quickly that more than one chapter has to be scanned in order to find the correct set up. To be precise: Chapters 3, 4, 7 and 8 had to be consulted when we set up our example.

After preparing the data the next stage is the specification of the multilevel model. This procedure includes listing all possible variables that may be used in the sequence of the different models that are to be fitted. In our example we use only one interaction: between a micro and a macro variable (the only two variables in our data set), which means we fit four different models: a random intercept only (Model 1); adding a random slope, no second level variable (Model 2); adding a second level variable interacting with the intercept only (Model 3); and second level variable interacting with both intercept and slope (Model 4).

Model 1: random intercept only, no second level variables, no interaction (see the exclusion of these variables in the first statement)

```
FPAR "OTL" "XOTL"          (see Chapter 7 for this and following commands)
SETV  1   "CONS"
SETV   2    "CONS"
START                           (see Chapter 8)
```

The second run is again a matter of one or two statements, a process that is easier and faster than going through a whole sequence of questions (compare VARCL and HLM).

Model 2: as efore, but a random slope is added:

```
    CLRV 2                      (this statement "wipes the slate clean," or delete all or
                                     part of the requests given in the model fitted
before,                                   see Chapter 7)
    SETV   2   "CONS"   "X"     (two random coefficients: intercept and X, see
                                Chapter 7)
    START                       (see Chapter 8)
```

Model 3: adding an second level variable, OTL, that only interacts with the intercept:

```
CLRV   1
CLRV   2
FPAR   "OTL"                (F is for fixed part)
SETV   1   "CONS"
SETV   2   "CONS"
START
```

Model 4: as before but an interaction, OTL (second level variable), is added with the slope:

```
CLRV   2
SETV   2   "CONS"   "X"
START
```

This is the kind of simple example that we missed in the manual. By scanning back and forth between Chapters 7 and 8 we were able to fit one model next to the other.

The complexity of the manual is a result of the following nice feature of ML 2: ML 2 is integrated with the general statistical package NANOSTAT. Many commands are available to modify data before starting the multilevel analysis; for example:

1. Creating, recoding and transforming variables.
2. Sorting and selecting cases. (In order to use ML 2, as well as all other packages, the data must

be ordered, so that all cases belonging to a given level are sorted together.)

3. Deleting cases (listwise) with missing values.
4. Conducting exploratory plotting.
5. Computing descriptive statistics and performing simple analyses such as OLS regression.

Since many commands are available, a HELP program is built in. This online help facility shows the commands and/or the commands' format.

Of all the programs mentioned in this report, ML 2 is the one that allows the users the most freedom to choose input and even what happens during the run. From the moment that the model is specified and choices are made, to the time the program starts running, ML 2 allows control. After the command STAR (the start of the estimation process), the user can speed up the procedure by typing in NEXT (a request for the next iteration) or stop the program by typing in FREE (to freeze some estimates), followed by the name and column of the variable that should be stopped from being estimated any further. When the user sees from the iteration process that some estimates are unstable, the convergence can be hastened by holding the instable parameters constant at their most recent estimated values and continue with fewer parameters. FIXE and RAND commands also can be used after convergence has been reached or at the end of any iteration. By typing FIXE, the IGLS or RIGLS estimates of the fixed coefficients and their standard errors are obtained; by typing RAND, the same happens for the random parameter estimates. Together with the random estimates the dispersion matrix is obtained. The default output is simple, without extra and maybe unnecessary information.

To make use of the full potential of the program, extensive experience (or going to one of the many workshops offered by the program authors) is necessary. The reward for the user is that the program obviates the necessity of preparing the data in advance in another package. Another advantage is that it is easy to make adjustments or new interactions in a later modeling stage.

The default output is limited. Only the essential parameters and the respective standard errors are provided. It is possible to choose an output that, for example, includes OLS Estimates or RESIduals to test CONTrast. Missing in the output are posterior means for intercepts. Slopes are provided that are not in the output, although they can be computed quite easily from the "shrunken" residuals.

H. Special Features.

The RESE command is an on/off option that allows pairs of random coefficients to have a nonzero covariance when one of thr pair has a variance of zero. This option is useful in explicit modeling of variation when one wants to avoid constraints inherent in particular parametrization. See Goldstein (1987, p. 36) for an example.

The BATC command switches the mode of operation between batch (no pauses) and interrupted (pauses between iterations). The pause option is useful when starting a new analysis to get a sense how the estimates change from iteration to iteration, which indicates which parameters have stable estimates and which do not.

The option to enter starting values for the parameter estimates other than the default OLS estimation is a special feature. These initial estimates can be read into the appropriate columns using the READ command. The manual explains how to read these data by using either a file or by manual entry. This information is found in the part that explains the use of the larger software package into which ML 2 is built (see Subsection 4.1.7 of the manual). The program can analyze two levels. (ML 3 is available but it is not quite finished.)

Another feature in this software package lets the user model a simple multilevel logit and log-linear model. This allows the researcher to analyze survey data with proportions or binary variables as the dependent variable. A simple multilevel logit and log-linear model is described in the manual. ML 2 also can include a variable in the random part that is not included in the fixed part, a feature that program shares with HLM. The manual explains the value of this option in the section dealing with a logit model (see page 23).

Information about the convergence process is provided; this is useful since it is possible in ML 2 to interrupt the program and "freeze" the estimation of individual parameters for the rest of the estimation during the run. This is especially helpful in situations where some parameter estimations seem to converge slowly because the estimated variance of those random parameters are close to zero. The freezing of the estimates speeds up the convergence. ML 2 alerts the user to slow converging parameters during the session, and the user can take action by freezing those parameters instead of waiting too long for convergence.

ML 2 provides predicted values for level 1 and 2 residuals and their comparative and diagnostic standard errors. These values can be used for several purposes; for instance, the user can obtain the estimate of the random coefficients for each level, or she can check the model assumptions. Useful suggestions can be found in the user's guide.

Very special is the fact—mentioned several times—that ML 2 is built into or around an *existing software package*. Although some programs (HLM, GENMOD) allow preliminary data analysis to explore the relation in the data before the hierarchical model is fit, this package is unique in the variety it offers in this respect. Simple commands are available to: create, recode, or transform variables; sort and select cases and delete missing data; conduct exploratory plotting; and to compute descriptive statistics and perform simple analyses. This is very useful feature since it solves the problem of opening and closing various programs and of transporting data whenever a multi-level analysis is performed after data modification.

ML 2 offers many possibilities for the advanced user of two-level analysis techniques. For the same reason it is not an easy program to start out with.

2.4. VARCL

A. Design Philosophy

VARCL was initiated by Aitkin and Longford (1986) and produced and maintained by Longford. A random coefficient type of analysis, it can be used to analyze multilevel data. It provides the option to fit random slopes, but has no possibilities for interactions between slopes and second or third level variables. VARCL is a single purpose package, designed for the fitting of mixed linear models with nested random effects to data involving hierarchies of nesting. It consists of two independent computer programs, VARCL3 and VARCL9. VARCL3 is used for the analysis of data with three or two levels of nesting and for studies in which modeling of the variation of the within-area and within-group relationships is of interest. VARCL9 can be used for analysis of data with up to 9 levels of nesting, but it permits only simple structure of the random effects. There is no requirement for the balance of the nesting structure in either program.

B. Implementation details.

Both programs are written in FORTRAN 77 and require an interactive computing environment. VARCL was originally written for VAX/VMS, but its has been ported successfully to PC/DOS, MacOS, and Sun/Unix environments. VARCL3 is complex and has a more elaborate interface than VARCL9, but the two are similar enough to warrant a single user's guide. The interface of VARCL combines interactive and batch features. The batch feature is the control file that contains declarations related to the data set such as the title, data file names (the data set may consist of several data files), formats, variable names, nesting structure, etc. Having this information available in a separate file makes the interactive session less tedious.

The implementation restrictions are defined in the include file IMPLE.ADD, thus they can be changed very easily by recompilation. There is no limit on the maximum number of first level cases. With the DOS version we have worked with a maximum number of variables equal to 24, a maximum number of factors of 24, and a maximum number of sufficient statistics equal to 30.000.

For VMS and for a Mac with 5Mb RAM these limits can be increased without problems to 48 / 48 / 300000. The number of factors and of variables, as well as the number of degrees of freedom, must not exceed the number in the IMPLE.ADD file, which is echoed in the output header.

C. Models

The models fitted by VARCL generally are quite different from the ones fitted by GENMOD, HLM, and ML2. VARCL3 fits the following maximal model (in the two-level case)

$$\text{(VARCL1):}\quad \underline{y}_{ijh} = \sum_{k=1}^{K} \underline{b}_{jhk} x_{ijhk} + \underline{\varepsilon}_{ijh}$$

The random coefficients are further specified by

$$\text{(VARCL2):}\quad \underline{b}_{jhk} = \gamma_k + \underline{\delta}_{hk} + \underline{\psi}_{jhk}$$

The disturbances satisfy

$$\text{(VARCL3):}\quad \underline{\varepsilon}_{ijh}\ '\ \mathrm{iid}\ N(0, \sigma^2),\ \underline{\psi}_{jh}\ '\ \mathrm{iid}\ N(0, \Omega_2),\ \underline{\delta}_h\ '\ \mathrm{iid}\ N(0, \Omega_3)$$

Here the matrices Ω_2 and Ω_3 are of order K.

Various restrictions are possible within the framework of the maximal model. The $\underline{b}_{jhk}$ can be restricted to be equal to zero (a variable does not occur); we can require $\gamma_k = E(\underline{b}_{jhk})$ to be equal to a given constant (for instance zero, then the variable only has a random part); we can require one or both of the random components to be zero or nonzero (in four possible combinations). The variances (diagonal elements of Ω_2 and Ω_3) can be restricted to be equal to given positive constants (if they are nonzero). In fact, all individual elements of the covariance matrices can be restricted as well. Because of invariance considerations, it is suggested that covariances between intercepts and slopes be left free. It is clear that a large variety of models can be created by using these restrictions, although nontrivial first level random effects (as in ML2) are not possible.

In VARCL9 we have an inherently more simple structure for the error terms. For four levels, for instance, the general model is

$$\text{(VARCL4):}\quad \underline{y}_{ijhg} = \mu_{ijhg} + \underline{\delta}_{g} + \underline{\delta}_{hg} + \underline{\delta}_{jhg} + \underline{\varepsilon}_{ijhg}$$

Here μ_{ijhg} is the fixed part of the model, and each of the disturbance terms is a single normal random variable, independent from anything else. Thus, there are no covariance components and no random slopes. The extra parameters (next to those in the fixed part) are L variance components, if there are L levels.

D. Routines.

For his VARCL program, Longford (1987) uses the Fisher scoring algorithm. The manual describes the algorithm in detail (quite unlike the black box approach in the ML2 manual). Two possible complications can occur during the iterations of the scoring algorithm. Both are dealt with explicitly and automatically by VARCL. If one of the estimated dispersion matrices becomes indefinite (has negative eigenvalues) during the iterations, the iterations are *damped*. Thus the program does not go all the way, but forces positive definiteness by interpolating. This may result in a considerable deterioration of the convergence behavior of the algorithm. (More elegant solutions to this problem are possible—one is updating a triangular factorization of the dispersions; see Lindstrom & Bates, 1988.) The program prints the message that a *covariance adjustment* has taken place.

The second problem occurs when the information matrix, used in the scoring iterations, becomes singular. The offending parameter is then *aliased* (i.e., excluded from the model). Aliasing obviously improves the convergence, but results in fitting a different model. It is irreversible (i.e., once a parameter is aliased, it will not be unaliased and left free to vary anymore). It has been our experience with VARCL that aliasing occurs in situations with complicated models, in which the EM algorithms of GENMOD and HLM exhibit very slow convergence. In the case of aliasing and in the case of covariance adjustment, the VARCL manual suggests fitting a smaller model.

E. Data Setup and Data Handling

The input data for the outcome and explanatory variables can be stored in separate locations, one location for each level. For instance, files can be referred to as subject-level, group-level, area-level, and so on. The subject level (level 1) is also referred to as the "elementary" level. The data can be stored in separate files with fixed F-formats; the control file with the basic information has to be written in a special way specified in the manual. The data can also be stored in a single file (group level and individual level together), or in two or three separate files as long as the information is not mixed between the three files. The data matrix has to have a hierarchical ordering, which means that observations belonging to the same context or group have to be grouped together. The basic information (a batch job provided by the user) has to contain the number of observations per group in the order they appear in the data matrix because no facility is provided by the program to read a group identification.

Basically VARCL does not fit multilevel models, but hierarchical random coefficient models. It accomodates random slopes and an interaction between group-level characteristics and intercepts, but not between slopes and group level variables. By using an input matrix with specially created interaction variables, the program can be used in the same way as the other programs are. Here again, no missing data can be handled in the model fitting stage.

A session with VARCL consists of the following steps:

1. A declaration of the maximal model. Declaration of the maximal model is fully interactive and is made only once in each session. A maximal model is declared by answering three prompts: (a) a declaration of the y-variate and sampling weights (if there are any weights, and only variables defined on the elementary level are acceptable as response variables); (b) a declaration of the fixed part of the model; and (c) a declaration of the random part of the model. This third declaration is only relevant for VARCL3, because only models with simple random parts can be fitted in VARCL9.

2. After declaring the maximal model, the program proceeds to input the data (as declared in the control file) and to compute the sufficient statistics.

3. Fitting of new models. VARCL allows interactive fitting of several models to a data set. Before fitting the first model, the user has to define a maximal model that contains all the models considered for fitting as its submodels. Only variables declared in this step, which is the first step above, can be used in the following steps. This is the most important part of the session. In later steps, fitting of new models is restricted to these variables (in still later steps, the fitting of models is restricted to subsets of these variables). Each model fitting step consists of a new cycle of model selection (carried out interactively by the user), model fitting, and display of the results. This means that after the calculation of a set of sufficient statistics for a model, the user is presented with a default model for fitting. Alterations in this model can be made followed by the fitting of the approved model. Next, the user can inspect the results, make further alterations, and so on, consecutively fitting any number of models in a session. Any model can be selected as long as it is a submodel of the maximum model declared in the second step. Usually a data set is analyzed during several sessions of VARCL, with different maximal models declared in each session.

F. Results

Output: A session of VARCL can be saved in a binary "dump" file which contains the entire information required to carry on, in a new session of VARCL, where the old session was terminated. If the user wishes to restore such a dump file, the answer should be "Y" to the prompt

```
>WANT TO RESTORE A DUMP FILE ?
```

This is followed by the prompt that asks for the name of the dump file. The basic information, the maximal model, the sample means and proportions, and the results of the last model fitted are stored. The dump files can only be used for data with normally distributed error terms.

If the user does not want to restore a dump file, a valid name for another store file has to be assigned. This output file will contain the results of the analysis and a summary of the initial specifications. If no name or an unacceptable name is entered, the output will be directed to the terminal only. This is a non-fatal error, and the session will continue after the message

```
>OUTPUT DIRECTED TO THE TERMINAL.
```

Several models as output: The interactive fitting of several models within the maximum model allows the researcher to compare the fit of these models with each other or with the maximum model. By using the deviance from the maximal model and comparing this with the deviance of other models and the difference in degrees of freedom, it is possible to make a choice between one model and another that does not fit as well, but is simpler. It is unfortunate that the VARCL manual does not provide guidelines for how to use the deviances in testing the differences in goodness of fit for the models fitted in one single session.

Several data sets are provided with the program. Two of them are used in the manual as examples.

The first one is the Gray et al. data set, a school effectiveness example. It contains data for 907 students in 18 schools. Sex and test scores are the main variables. The manual shows some model fitting with this data set. (See also Aitkin and Longford, 1986, for the use of these data.)

Next is a cherry tree example, with one level of observation, 30 observations, and 3 variables: diameter, height and volume.

Next is an example (The lower class men short forms academic profile, Fall, 1987) with three levels of observation: a subscore, the student and college level. The number of subscores is 18.320 (four observations per student over two variables). The number of students is 4580 and the number of colleges 32. Two variables at the subscore level and one variable at the student level are measured.

Another example, with five levels of nesting, contains 242 individuals nested within 49 groups in 14 areas within 5 states in 3 countries. Since we are dealing with simple models (only random intercepts are allowed in models with more than 5 levels), only two individual variables are used. No variables are specified for higher levels of aggregation. This data set is used also as an example in the manual and is worked out for several models shown in the manual.

G. User friendliness.

The VARCL manual is, for the most part, user-friendly. The manual contains much valuable background information concerning the output and interpretation. Not so helpful are the output examples given without comments (one run with VARCL provides you with the same output). For the preparation of the batch job (the file where all the basic information is stored) some work has to be done, since for each group the number of observations has to be specified. This is a tedious job, especially when the number of groups in the data set is large. Extra preparation is also needed when interactions between first level and second level variables are of interest for the researcher.

The model fitting part of this program is very user-friendly and easy. It is possible to fit a large number of models within the declared maximum model in a single session. After each calculation of a set of sufficient statistics, the user is presented with a default (only random intercept) model for the next model fitting stage. The program fits the chosen model by making alterations in this default model. After inspecting the last results, new alterations can be made, which are calculated, and so on, fitting consecutively any number of models in one single session in an easy way. Although model fitting involves the bulk of the computational load, the user is spared repeated input of the data. Compared to the other three software programs, this is a very nice and also unique feature. In general VARCL is quite friendly, but it is difficult to use mindlessly. The user has to know quite well which models to fit, which interaction variables to create before starting any session, and which covariances to fit.

The speed of the convergence is another nice feature. Comparable programs take much longer to do the same job (we deal with this more extensively in Chapter 3). The algorithm is certainly fast scoring.

Also helpful are the error messages and the options to correct them. At the stage when the declared files are opened (but not read yet), the interactive part of the program can give several error messages when something is wrong and, at the same time, the option to correct these mistakes. Examples:

```
>REFUSAL. FILE CANNOT BE OPENED.
```

or

```
>FAILURE TO READ NAME AND TYPE OF VARIABLE NO:
```

Some errors are fatal, which means the VARCL session is terminated. This feature permits the user to check the data filenames or rectify the problems by other means.

The time and effort saving device at the variables declaration stage (especially when the number of variables is large) is nice. By choosing the option "(almost) all variables in the model," time is saved, because only the variables left out have to be declared. In cases where only a few out of many variables are used for the next model, the option "explicit declaration" is a quicker way to proceed. A third option is "Pick the variables for the model." This is the most secure choice, but it is also the more time consuming one. Here all variables are listed one by one and the user has to respond to the prompts by "Y" or "N." Unfriendly is the fact that the user has to remember (or make a note) of the number the program assigns to each variable (the intercept is number 1 and each following variable number 2,3,4,...,m+1), since the program only accepts numbers.

Helpful are the built-in checks and opportunities to correct mistakes in the just declared model. The maximal model (or any following model) is displayed again after the declaration in a table stating the Y-variate, prior weights, for each variable whether it is included in or excluded from the fixed or random part. This display is followed by the prompt

```
>ANY ALTERATIONS ?
```

If the user responds "Y" all the declarations of the maximal model made so far are invalidated and the interface returns to the declaration of the y-variate and weights. The same rules and defaults apply as described in the fixed part for the random part in the VARCL3 program, with one exception: only variables included in the fixed part are eligible for the random part.

H. Special features.

Unrestricted refit facilities. VARCL allows interactive fitting of several models to a data set. Before fitting the first model the user has to define a maximal model which contains all the models considered for fitting as its submodels. After declaring the maximal model the program proceeds to input the data, calculate a set of sufficient statistics, and present the user with a default model for fitting. The user can make alterations in this model, make further alterations, and so on, fitting consecutively any number of models in a session. This feature is so special because it allows the user to find conveniently the most parsimonious model by comparing models against the best fitting, but unrestricted maximum model. By doing so, it raises the user's awareness of the importance of the goodness of fit, instead of overemphasizing significant (separate) effects of (separate) coefficients and/or theoretical importance of these effects.

Unique is a *quasi likelihood adaptation* for non-normal (binary, binomial, Poisson, and Gamma-distributed) outcomes. The first page of the program output is immediately followed by the prompt relating to the type of error distribution. The choices are: normal error, binary or binomial error, Poisson error, or gamma error. It is clear that when the dependent variable consists of proportions or probabilities instead of numerical values, these options for the error distribution are more appropriate. For example, if a binomial or binary distribution has been selected, the user is asked for the binomial denominator with the prompt

```
>THE BINOMIAL DENOMINATOR IS VARIABLE NO:
```

The user has to respond with a variable number. For instance, for binary data the answer would be 1 (In VARCL the first variable is always the Grand Mean). For the Gamma error distribution the user is asked for the scale with the prompt

```
>ENTER THE GAMMA SCALE:
```

A positive number should be entered; most common again is the number 1. This last option is also available in the software package ML2 (see par. 2.5). Following these steps will automatically result in different estimation procedures. VARCL is the only program that allows the user to choose among four different error distributions. ML2 also incorporates a logistic option. HLM and CENMOD assume normally distributed errors.

The estimation of a covariance structure with three level data. The maximum number of levels of nesting is three for VARCL3, with full flexibility for modeling of variation. The maximum number of levels of nesting is nine for VARCL9, with simple variance component structure. In the three-level model only variables defined at the first level can be included in the random part. Variables definerd at the second or third level can only be included in the fixed part of the model. VARCL9 only decomposes variance at the several levels. No variables at any level interact with each other or have random parts.

A unique feature is the *option to declare a covariance structure*. The user may declare the covariance structure according to his/her own expectations. The extreme choices are: intercept by slope covariance only or (the other extreme) all covariances. Choices between these extremes are possible as well. When the first choice is "intercept by slope covariances only," the user is prompted to declare additional covariances. It is obvious from the assumptions of the theory behind the model that only variables included in the random part are allowed to have covariances with variables in the random part on the same level. This includes categorical variables with more than two categories. A categorical variable can be declared to have covariances with itself. Covariances will be declared for the pair of different categories other than the first reference category.

VARCL provides a special treatment for *categorical independent variables*. Categorical data, when declared as such, are changed into dummy variables. For each categorical variable m-1 (m is the number of categories), parameters are estimated. This feature is not of much use when using a categorical variable for interaction with another level variable, since the dummies have to be constructed anyway in order to make interactions with other variables before starting the analysis.

2.2. Summary

The usual restrictions imposed by the four programs we have discussed are:

1. All variables in the random part are included in the fixed part.
2. All level one coefficients are random at level two (the full random coefficient model).
3. The only explanatory variable whose coefficients could be considered to be random at level one is the intercept.

All four programs have ways to deal with the first two restrictions, and all leave room for variables that are not included in the fixed part to be random (HLM, ML 2), or to fit a mixed model (all programs), but the last restriction is only overcome in ML 2. On page 17 of the ML 2 manual an example is given in which a predictor of the first level has a random part at that same first level.

A test of significant improvement of fit between two models, a simple and a more complicated model for instance, is provided only in HLM. For the other packages, except for ML 2, it is easy to calculate the test statistic by subtracting the deviances of two separate models. In GENMOD, however, the likelihood does not seem to be computed correctly. Deviances can be used especially effectively when using VARCL with its "interactive fitting of several models" facility. The user can make alterations in the first fitted model, make further alterations, and so on, fitting consecutively any number of models in a single session. This feature allows the user to find conveniently the most parsimonious model by comparing models against the best fitting, but unrestricted maximum model. By using the deviance from the maximal model of more than one solution and comparing the difference with the degrees of freedom gained by estimating a smaller number of parameters, it is possible to make a choice between one model and another. Since the maximal model has the most parameters, by definition it has the best fit. However, the question is whether it is much better than a simpler, more useful (and easier to interpret) model. The deviance statistic can be employed in a likelihood ratio test by comparing the deviance of two or more models. This difference between the two statistics has a chi-square distribution. The chi-square test of testing the significance of the differences in deviance, in relation to the gain in degrees of freedom, has to be calculated by the user herself in order to compare the fit of the different models.

By doing so, it raises the user's awareness of the importance of the goodness of fit. In HLM the deviance of the tested model from the "full model" (the name of the maximum model) with the associated degrees of freedom and chi-square statistic are part of the output. Since HLM allows only one model next to the full model in each session, this output can be given. Because the maximum model can be followed by any number of different models in VARCL, and as a result the number of comparisons can be extensive, it is impossible to give those statistics in the output. It is unfortunate that the VARCL manual does not provide guidelines for how to use the deviances in testing the differences in goodness of fit of the different models fitted in one session.

The number of iterations is variable in all packages and is left to the decision of the user. The suggested number in most packages is from 10 to 15 iterations. In this paragraph we will show that for EM algorithms this number is definitely too small to get results comparable to those of other programs. Comparing HLM and GENMOD with ML 2 and VARCL in Table 2a,b and 4a,b, for instance, shows this clearly. The latter two packages, with fast linear or superlinear convergence, stay within the limit of 15 iterations, while the other two exceed that number considerably to reach the same convergence criterion. In our examples, more complicated models (more complicated than a model with only a random intercept and no second level variables) need many more iterations in the packages that use the EM algorithm (GENMOD and HLM) than in the ones that use scoring (VARCL) or weighted least squares (ML 2).

The (default) output given by the four packages differ from one page to several and from many parameters and significance tests to only the essentials. The default for ML 2 gives only the fixed and random parameters with the respective standard errors and the number of iterations. This may be nice for new users because they are not confronted with an abundance of output, within which the relevant numbers may be hard to find. The more experienced user needs more output, which can be provided on request, such as OLS estimates and residuals. In this respect GENMOD resembles ML 2, and both differ considerably from VARCL and HLM.

T-test values for parameters are provided only in HLM. This is not unusual for this type of software, since separate t-tests do not take the total analysis into account—this may lead to type-I errors when making interpretations from single t-tests. In addition, tests are more heavily dependent on statistical assumptions, such as normal distributions, than parameter estimates are. When we compared HLM to the other packages, we found that HLM provides a maximum of information in order to minimize the effort for the user. For example: Next to the usual estimates of the fixed part of the coefficients (gammas) and their standard errors, the t-statistics and p-values are also given. In contrast, the other three programs provide only the first two and leave it to the user to calculate the significance tests when this calculation is a necessary and/or responsible thing to do. The overall test (test of differences between deviances for the goodness of fit) may be more reliable here.

Packages differ in the way they handle the raw data set. Centering is a much discussed issue in recent studies (Raudenbush, 1989a,b; Longford, 1989b; Plewis, 1989). One of the options is to center variables around group mean or grand mean. An option offered in ML 2 is the choice to center variables used in the fixed and random part around the grand mean in either, both, or none of the parts with the options FMEA (fixed part) and RMEA (random part). The reason for centering, as given in the ML 2 manual (page 14), is that centering sometimes facilitates interpretation, but that it is used mainly as a way to improve the numerical performance of the estimation algorithm. Variables are replaced by their centered version in VARCL to make computation easier. The deviation is a deviation of the grand mean, not of the group mean. No options are available to change this default centering in order to use raw scores. But since the outcome is reparametrized in terms of the original data, this leads to the same outcomes produced by programs that use raw data (as shown in the comparisons in Chapter 3). This also leads to the conclusion that centering in VARCL is of no consequence. In HLM the authors have built in a question in their interface: "Do you want to center variables?" If the answer is "yes," the user must introduce the names of the variables to be centered. The choice to center none, all, one, or more variables is left to the user. This is, however, centering around the group mean. Centering around the overall mean is not an option and has to be done before starting HLM. The manual explains that this type of centering may be advisable to make interpretation of the intercept easier. Centering around the group mean is, in principle, fitting another model. We do not use this option on the examples in this report for this reason. We will return to this topic briefly when discussing

Table 2c in the next chapter.

Information about the convergence process and the use of this information differs significantly among the four programs. In the program ML 2 it is possible to interrupt the run and *freeze* the estimation of individual parameters for the rest of the estimation during the run. This is helpful in situations where some parameter estimations seem to converge slowly because the estimated variance of those random parameters is close to zero. The freezing of the estimates speeds up the convergence. The user of ML 2 is informed during the session of slow converging parameters and can take action by freezing them. A somewhat similar concern is formulated in the manual of HLM, but the user is not allowed to take action during the run. Rather, the user is informed afterwards when she looks at the chi-square table and the deviance statistics of the run. By setting the residual parameter variance at zero in the next run for those variables that slowly converged, the same effect is reached in HLM as is reached in ML 2 by *freezing* during the run. These boundaries are handled in various ways. Programs using the EM method need no special provisions to deal with boundary constraints. Parameters can never get outside, variances are never negative, and matrices are never exactly singular. Nevertheless, EM methods that converge to boundary points generally have sublinear convergence (Horng, 1987). It is not entirely clear that this boundary is treated efficiently in ML 2 and VARCL. What happens if a matrix is almost singular? Will the parameter be set to zero (saves time), or will the iterative process merely slow down? Differences among programs (or versions of the same program) may result in different solutions (see Chapter 3).

Hypothesis tests are possible within some packages. In ML 2 the residuals can be used in two different ways. Firstly, the level two residuals may be, and often are, used directly to provide estimates of the unknown level two effects (this procedure is beautifully explained in Aitkin and Longford, 1986). ML 2 will provide estimates of the standard errors of these estimates, so that confidence intervals may be examined. When sample size is large, the estimates will have an approximately normal distribution, and confidence intervals can be estimated in a reliable way. Secondly, ML 2 can calculate simultaneous confidence intervals for a set of contrasts that the user specifies as well as the confidence interval for each contrast considered individually. Thus, the user is able to test complex hypotheses about several elements of the matrix gamma. By using a contrast matrix (equivalent to the matrices used in the Scheffé test for effects in ANOVA), several combinations of effects can be tested against each other without enlarging the alpha level. Hypothesis testing is also offered as part of the HLM program, but in a slightly different way, basically to univariate t and multivariate T testing.

The use of interaction variables, such as the interaction of a (fixed) macro level variable with a random micro level variable, is quite common in multilevel data analysis. But not all packages provide equal opportunities to use interaction variables. It is relatively easy to use these types of interactions in the programs GENMOD and HLM. Use is less obvious in ML 2 and almost forbidding in VARCL. In the ML 2 manual this is emphasized in the first example. It is clearly indicated that interaction variables (interactions between first and second level) have to be created before the analysis starts. It can easily be done in "the specification of the model" phase by using the command MULT, followed by the two names of the variables and a position for the new created variable (format statement). In the VARCL manual no mention is made of the fitting of interactions between random first level and fixed second level variables. Basically, VARCL does not fit random coefficient models, but variance component models. It permits random slopes and an interaction between group-level characteristics and intercepts, but not interactions between slopes and group level variables. This means that VARCL does not create variables during the session, as do HLM and GENMOD, nor does the program allow the researcher to make these interactions as does ML 2. An earlier version of VARCL did allow the user to make interactions, but this feature was removed from later versions. (Also removed was a module to include various transformations of the variables.) It is, however, possible to fit a model with interactions by fooling the program and using an input matrix with specially created interaction variables. By treating these interactions as fixed micro level variables in the model, the same results can be obtained with VARCL as are obtained with other packages, as will be shown Chapter 3.

Some packages allow dichotomous dependent variables. A specialty offered in ML 2 is the opportunity to model a simple multilevel logit and log-linear model. This allows the researcher to analyze survey data with proportions or binary variables as the dependent variable. A simple multilevel logit and log-linear model is descrribed in the manual. This feature is shared with Longford's program VARCL, which also includes an option for the error distributions to be either logistic, logarithmic, or reciprocal. We understand that there is also a version of GENMOD to deal with logistic multilevel models, but we have not seen it.

	GENMOD	HLM	M L2	VARCL3
availibility	shareware, from author; source included	commercial, from software house, no source	commericial, from author, no source	shareware, from author; source available
literature references in manual to basic literature or applications of the hierarchical linear model	few, mainly to theoretical or basic work	many, mainly to applied work	many, mainly to applied work	few, mainly to theoretical or basic work
ease of use of the program and manual	the manual is difficult to read and the program set up complicated	the manual is clear [illegible] program easy [illegible] No program set-up, but answers to questions.	the manual is clear, but it is hard to find the program set-up for a particular problem, due to the many choices	the manual is not easy but the program is easy to use. No program set-up, but an interactive answer and question system.
estimation procedures	restricted maximum Likelihood REML	restricted maximum likelihood REML	Generalized Least Squares, restricted or unrestricted IGLS and RIGLS	Full Information Maximum Likelihood FIML
possibilities within the package for data manipulation before or after the modelling stage	none, the data has to be prepared in advance	a lot; before, within, and after; see the manual	a lot; data exploration and preparation before and after modelling: plots of residuals etc.	none, the data has to be fully prepared in advance
interface	batch job	fully interactive. The program asks questions, the user answers.	interactive, user gives a stateme[illegible] programs answers	batchjob for data declaration. The run stage is fully interactive: the program asks the user answers
complication of the preparation of the dataset	identification for both levels. Data have to be hierarchically ordered	identification and hierarchical ordering of the data	no preparation, all the necessary preparation (interactions and ID) can be done within the package	preparations of the datset can be complicated; necessary interactions prepared in advance
default output	in the batch job many parameters are available to ask for a lot or only the essential parameters	default output is a lot, but it can be expanded with more tests	output can be regulated from very essential to very much, by way of the commands used in the interactive stage	output is the whole history. Only the posterior means and the iteration history can be surpressed
Specialities and/or unique features	allows variation over context of residual variance. Allows different models over diff[illegible]nt contexts	hypothesis testing (shared with GENMOD) The estimation of reliability coefficients	The choice between restricted and unrestricted estimation. Proportions as the dependent variables.	Three levels and up till nine levels for simple models. Allows for proportions as the dependent variable.

	GENMOD	HLM	M L2	VARCL3
options for weighting	no	yes	no	yes
variance-covariance adjustments	not necessary, the EM algorithm	not necessary: the EM-algorithm	???	aliasing and covariance adjustments.
small data sets (# observations within groups less than # variables)	no	yes	yes	yes
performance	very slow convergence	faulty singularity test	fails for some small datasets in RGLS	adjusts covariances
documentation	not good	good	good	average
ease of learning	hard	very easy	very hard	easy
ease of use	moderate	easy	very easy	easy
error handling	good	moderate	moderate	moderate
speed	slow	fast/slow	not fast	very fast

3. COMPARISON OF THE PROGRAMS

3.1 Introduction

In this section we compare the various programs. The comparisons are far from complete, for various reasons. We briefly discuss those reasons here.

Some programs are not finished products (although ML2 and HLM seem to be approaching this stage). During our experiments we discovered bugs in all four programs. We reported these problems to the authors, and the problems were (or will be) corrected. We think that all four programs may still contain bugs of some sort, because all four give strange output under some circumstances. As far as ML2 is concerned, we do not have the source code for this program, and we cannot really experiment with it in the same sense that we can experiment with the other programs. It is difficult to decide which version of ML 2 to use. While we were printing the first version of this report, we received a new version of ML2, which was nearly three times faster than the original version. During the first and second version of this report, HLM was updated to version 2.1 and VARCL was modified. This is somewhat frustrating, but we have to live with it. We have tried to address the changes made by the authors as much as possible, but at some point we obviously had to stop. Changes reported after November 15, 1989, have not been incorporated in our comparisons.

Secondly, comparisons are difficult because the programs are different in various unfortunate aspects (at least for our purposes). ML2 does not write out the value of the likelihood function or the deviance, HLM writes the value of the restricted log-likelihood, VARCL the value of the unrestricted deviance, and GENMOD writes out both values (but minimizes only the first one; moreover, it seems to write out the wrong value). For large examples GENMOD does not give sufficient precision in the output to compare values of the likelihood function with those of other programs (because the authors want to have the output for each iteration on a single 80-column line). The stopping criteria for each program are very different. ML2 and VARCL have fast linear convergence; in fact, the convergence is actually close to superlinear if the model fits well. GENMOD and HLM typically have slow linear convergence, and the default stopping criteria for GENMOD are much more conservative than those of HLM. Thus, comparisons of convergence should really be in terms of the likelihood function, but we have already seen that this leads to unexpected difficulties.

Thirdly, programs may have restrictions which may not have been intended by the authors. HLM refuses to perform at least some functions if a within-group cross product matrix is singular. GENMOD, in its original versions, had different individual-level error variances for each group, and consequently could not be compared with the other programs because of this. In the latest version the option of restricting all these error variances to be equal has been added. VARCL uses aliasing and covariance adjustment in case of singularities, which is perhaps a good idea, but which again makes comparisons very difficult.

Thus, we are forced to give only some preliminary comparisons, and we will continue to work with the authors of the programs (at least with the authors of VARCL, GENMOD, and HLM) to improve these comparisons. Most of our comparisons are on AT-type machines running DOS, although we also have versions of HLM for TSO, CMS, and UNIX; of VARCL for VMS, CMS, and the MacOS; and of GENMOD for UNIX and MTS (a local IBM mainframe OS). We will extend at least some of the comparisons to these other operating systems and machines. We will report some within-machine and some between-machine comparisons.

Comparing the same program which fits the same model on the same data on different machines seems simple, but is actually quite complicated for various reasons. In the first place, not all programs run on all machines. For ML2 we can only compare within the DOS/OS2 family, for instance. HLM runs on PCs and (at least on some) Unix machines. We have not been able to compile it (so far) under BSD 4.3 on the Sun or under A/UX on the Mac II. We will continue to try to do so. A Macintosh version of HLM has also been attempted, but without success so far. GENMOD runs on PCs with DOS, on Apollos with UNIX, and on IBM mainframes with MTS.

We have not obtained (and not tested) the UNIX version yet. A Macintosh version is being developed, but we have run into trouble here as well. Work on this version (done with Albert Anderson and Bill Mason) continues. VARCL was designed on the VAX, under VMS. Consequently, it is no surprise that it readily compiles and runs on VAX. With some minor adaptations, we have also produced versions for the Mac OS (using Language Systems FORTRAN under MPW), for 4.3 BSD (using f77 on the Sun 3/280), for A/UX (using Absoft RAT compiler for A/UX), for IBM 3090 under VMS, and for the various members of the PC family. If portability is a criterion, it seems that VARCL is the best choice, followed by GENMOD (which is designed for portability, but may have some bugs), then by HLM (which has different implementations for the PC and UNIX systems), and finally by ML2 (which is inherently non-portable because there is no source and because it is built into the NANOSTAT package). Of course, we ignore compiler effects in this comparison, although we could (and perhaps will) compare Lahey and MS-FORTRAN versions of VARCL and GENMOD. HLM has the main driver and some screen control functions written in C, which makes a Lahey version complicated to write.

Portability in itself may not necessarily be good. There is, for instance, the version of HLM that uses screen support functions on the PC (a first step in the direction of using windows), and the version with a termcap interface for UNIX. There is a version of ML2 that assumes that one has a coprocessor, and a version that does not assume this. Ideally, one would like a version of each program that uses the strong points of each machine/OS combination (for instance, the Toolbox on the Mac, MS-Windows on the PC, X Windows on UNIX or VMS machines with bitmapped screens, cursors on dumb terminals connected to UNIX computers, SPSF on terminals connected to IBM mainframes, and so on). These versions are no longer portable, but in order to build them with any efficiency, one needs a computational core that is as portable as possible.

We shall discuss some experiments that indicate why intermachine comparisons are complicated, even with a portable program such as VARCL. To make the comparisons we used a new version of VARCL, which writes the time of day (hours, minutes, seconds) at the start of each iteration. Real time measurement as soon as iterations start now seems reasonably reliable. However, in the interactive stages of the program they are rather useless, because we are to a large extent measuring our reaction speed.

3.2 Data, Models, and Results

We use six data sets for our comparisons—SIMS, GALO, GRAY, OSHEA, WEBB, and VOCAL—which we describe briefly before we give the results of the analyses.

3.2.1 SIMS

The SIMS (Second International Mathematics Study) data is taken from a national sample of United States eighth-grade students who took a series of mathematics achievement tests conducted by IEA (the International Association for the Evaluation of Educational Achievement) in 1981-1982. For this study, 3691 cases out of approximately 7500 were extracted. There are 190 school classes. Only two student-level variables, the sum of PRETEST core items and the GAIN score (difference between POSTTOT and PRETOT), are used. The second level variable is OTL (Opportunity to Learn).

The within-group model is (using a simplified version of the notation of chapter 1)

$$\underline{(GAIN)}_i = \underline{\beta}_{0j} + \underline{\beta}_{1j}(PRETOT)_i + \underline{\varepsilon}_i,$$

and the between-group model is

$$\underline{\beta}_{0j} = \gamma_{00} + \gamma_{01}(OTL)_j + \underline{\delta}_{0j},$$
$$\underline{\beta}_{1j} = \gamma_{10} + \gamma_{11}(OTL)_j + \underline{\delta}_{1j}.$$

The first version of the model has $\underline{\delta}_{1j} \equiv 0$ and $\gamma_{01} = \gamma_{11} = 0$ (a random intercept model); the second version has a random intercept and a random slope (and they are correlated), but still no second level variable (so $\gamma_{01} = \gamma_{11} = 0$). The third model has, in addition, the macro variable OTL.

The results of fitting these three models are in Tables 1a, 1b, and 1c.

Table 1a: SIMS data, random intercept model, no macro variable

	GENMOD	HLM	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	7.027	7.027	7.028	7.024	7.024
(gamma)10	-0.192	-0.192	-0.192	-0.1917	-0.1916
RandomPart					
(sigma)	22.52	22.52	22.52	22.51	22.51
(omega)00	9.443	9.443	9.442	9.374	9.374
Additional					
TotalIter	9	5	5	5	7
TimeIter	220	9	50	38	8
deviance		22306.6			22380.7

Table 1b: SIMS data, random slope model, no macro variable

	GENMOD	HLM	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	7.060	7.060	7.060	7.055	7.0553
(gamma)10	-0.186	-0.186	-0.186	-0.1857	-0.1858
RandomPart					
(sigma)	22.23	22.23	22.24	22.24	22.240
(omega)00	14.52	14.53	14.491	14.36	14.329
(omega)11	0.009	0.009	0.009	0.0088	0.00885
(omega)01	-.2342	-.2370	-0.2340	-0.2297	-0.229
Additional					
TotalIter	189	76	10	10	14
TimeIter	480	16	180	142	11
deviance		22382.4			22373.11

Table 1c: SIMS data, random slope model with OTL as macro variable

	GENMOD	HLM	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	0.0627383	0.069168	0.06191	0.03242	0.039131
(gamma)01	0.234197	0.234021	0.2342	0.2349	0.234701
(gamma)10	-0.228330	-0.229388	-0.2282	-0.2236	-0.224472
(gamma)11	0.0008590	0.000891	0.0008554	0.0007268	0.000751
RandomPart					
(sigma)	22.1318	22.1265	22.13	22.14	22.139398
(omega)00	12.65	12.68382	12.64	12.38	12.362
(omega)11	0.0119	0.01141	0.01117	0.01046	0.010
(omega)10	-0.2302	-0.23294	-0.2300	-0.2205	-0.220
Additional					
TotaIter	145	59	10	10	13
TimeIter	515	25	242	165	18
deviance		22367.8			22340.7

The rules for interpreting these tables (and the other ones in this chapter) are simple. The easiest procedure is to refer to the formulas for the corresponding model, which employ the same notation. The fixed regression coefficients (gammas) are given first, with (gamma)ij indicating the effect of macro variable j on the regression coefficient of micro variable i. Thus (gamma)00 is the fixed intercept, (gamma)i0 is the effect of the macro intercept on the micro variable i (i.e., the regression coefficient of micro variable i), and (gamma)0j is the effect of macro variable j on the micro intercept (i.e., the regression coefficient of the macro variable j). In the same way (gamma)ij with both i and j not equal to zero is the regression coefficient of an interactive variable, the product of micro variable i and macro variable j.

The second part of each table gives the random components. We use (sigma) for the variance of the first level disturbance. The variances and covariances of the random part of the regression coefficients are the (omegas). The (omega)ii are variances of the random slopes, with (omega)00 the variance of the random intercept. Also, (omega)i0 is the covariance between the randc m components of the intercept and the slope of variable i. In general, (ome, a)ij is the covariance between the disturbances of the regression coefficients of variables i and j.

In the last part of each table we give some additional information. TotaIter is the total number of iterations, TimeIter is the time per iteration. The product of the two is the total time spent in the iterative process. The total number of iterations depends, obviously, on the precision we impose. In the current version of this report some of this additional information is still missing.

Our general rule here is to use the defaults, adjusted in such a way that the programs seem to converge with about equal precision. Generally it is difficult to do this in a completely satisfactory way, for the reasons mentioned above. GENMOD has two stopping criteria: one at the absolute change and one at the relative change in the parameters. The others have as a stop criterion the absolute change in the parameter estimates. HLM has another stop criterion—it uses the (restricted) likelihood function. In some of the tables, but not all, we also give values of the likelihood function or the deviance at the optimum. Tables 1a and 1b are computed on a IBM/PC 80286 (6 Mhz); the results of Table 1c were computed on a much faster 80386 (20 Mhz) machine. This explains the smaller number of seconds needed for each package (TimeIter) per iteration. The IBM/PC 80286, although the slowest, is used throughout this chapter since it is the machine most commonly available to researchers. The time comparisons also include compiler and overlay effects.

Table 1a shows that the outcomes of comparable programs are the same up to two decimals. Remember that the convergence is not comparable in all cases because the programs sometimes use different stopping criteria. Another difference is that the first three programs use a restricted maximum likelihood method, while the last two use full maximum likelihood. This is also the difference between ML2(R) and ML2(F), since ML2 offers a choice between the two estimation procedures. The difference in the solutions that the different estimation methods produce (R or F) is clear in all tables in this chapter. The difference is more pronounced in small data sets than in large ones, and in complicated models with random slopes than in simple models with only a random intercept. Compare the solutions for RML and FML of the larger SIMS data set with the smaller GALO data reported in the next paragraph. These data sets are 3691 students in 190 schools and 1290 students in 37 schools, respectively. Also compare the solutions for the simple model in Table 1a with the more complicated model in Table 1b in this section and Table 2a with Table 2b in the next paragraph.

If we compare times for the 286 machine we see that VARCL and HLM are clearly the two fastest programs for simple models (with only a random intercept), as shown in Table 1a (and Tables 2a and 4a in the next paragraphs) The time needed to reach the convergence criterion in the more complicated models with random slopes, shown in Table 1b (and Table 2b and Table 4b in the next paragraphs), is many times larger (from four times more for GENMOD; as much as forty times more for HLM). VARCL is still by far the fastest, and GENMOD is by far the slowest, but the HLM program is fairly slow, slower than ML2(R) in most instances. The faster programs for fitting complicated models are the two that use FML (i.e., VARCL and ML2[F]). We have also run VARCL on the Mac II (Language Systems FORTRAN) and on the Sun 3/280 (FTP coprocessor, f77 compiler). On the Mac II each iteration for the random intercept model took 4 seconds; on the Sun 3 each took seconds (the last runs included some time to write to the terminal). The random slope model took 5 seconds per iterations on the Mac and 4 seconds on the Sun. The number of iterations and the solution were the same as those in Tables 1a and 1b.

3.2.2 GALO

These data were collected from 1290 students in 37 schools in the city of Groningen (see De Leeuw and Kreft, 1986, or Kreft and De Leeuw, 1989). For each pupil the individual level independent variables were gender (SEX), IQ, and occupational level of the father (SES). The dependent variable ADV represented teachers' advice on the most appropriate form of secondary education (scaled as a continuous variable). There was no macro variable.

The within-group model was

$$\underline{(ADV)}_i = \beta_{0j} + \beta_{1j}(SEX)_i + \beta_{2j}(IQ)_i + \beta_{3j}(SES)_i + \varepsilon_i,$$

and the between-group model

$$\beta_{0j} = \gamma_{00} + \delta_{0j},$$
$$\beta_{1j} = \gamma_{10} + \delta_{1j},$$
$$\beta_{2j} = \gamma_{20} + \delta_{2j},$$
$$\beta_{3j} = \gamma_{30} + \delta_{3j}.$$

The basic results for the random intercept are found in Table 2a; results for the random slope model are found in Table 2b.

Table 2a: GALO data, random intercept model

	GENMOD	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	-4.6738	-0.0325	-4.6740	-4.6770	-4.6766
(gamma)10	-0.0901	-0.0901	-0.0901	-0.0903	-0.0903
(gamma)20	0.0804	0.0804	0.0804	0.0805	0.0805
(gamma)30	0.1487	0.1486	0.1488	0.1489	0.1489
RandomPart					
(omega)	0.9074	0.9075	0.9075	0.9053	0.9053
(omega)00	0.0489	0.0488	0.0488	0.0466	0.0466
Additional					
TotalIter	11	11	3	4	5
TimeIter	12	2.5	34	19	3
deviance		3601.77			3569.61

* For HLM all variables were centered around their grand means.

Table 2b: GALO data, random slope model, no macro variables

	GENMOD	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	-4.5393	-4.5361	-4.5290	-4.5280	-4.6195
(gamma)10	-0.0600	-0.0605	-0.0592	-0.0605	-0.0686
(gamma)20	0.0783	7.8263*	0.0781	0.0781	0.0793
(gamma)30	0.1497	0.1499	0.1519	0.1520	0.1497
RandomPart					
(sigma)	0.8786	0.8793	0.8824	0.8818	0.8846
(omega)00	0.7391	0.7206	0.6569	0.5768	1.1230
(omega)11	0.0422	0.0402	0.0376	0.0339	0.0280
(omega)22	0.0002	1.5321*	0.0001	0.0001	0.0000
(omega)33	0.0011	0.0009	0.0000	0.0000	0.0000
Additional					
TotalIter	214	772	13	15	13
TimeIter	75	11	217	167	11
deviance		3578.36			3559.03

* Coefficients are based upon IQ/100.

These tables show more discrepancies in the solutions produced by the different programs (R or F) than we noticed in Table 1. Particularly evident are the discrepancies in Table 2b, where the more complicated model with random slopes is fitted. However, the deviant behavior of HLM with respect to the intercept in Table 2a is an artifact of our own data manipulation. To get HLM started we had to center all variables around the grand mean (centering of the independent variables only also would have done the trick, as we will discuss later). Use of the raw data set was not possible in this case and ended in an error message which contained the following:

```
At least some subset of the units must have sufficient data to
permit OLS estimation within groups to generate starting values.
```

```
One of the following problems or combination of problems has
occurred in every unit:
1. No variation in the outcome measure;
2. No variation in one or more of the within-unit explanatory
variables;
3. A singularity in within-unit data matrix for each unit.
We suggest that you carefully examine for each group the sums of
squares and cross products among the within unit variables.  If
these appear OK, try centering each of the within-unit explanatory
variables around the grand mean.  Since a condition check on X'X
is included, near singular but technically invertible matrices are
excluded from the starting value routine.  Centering around the
grand mean should solve this problem.
```

According to this message, all 37 groups were deemed to have singular cross-product matrices, and no data were left to start the iterations. The authors suggest that transforming predictors to deviations from the mean may solve the problem. It did indeed solve the problem, because HLM ran nicely after this transformation was performed. Centering of all the variables gave us the solution reported in Table 2a. Since centering in this case clearly violated the invariance considerations (see section 1.2.4.), we looked into it somewhat deeper. The results are given in Table 3. The problem, we found out, was not the data set, but the conditioning test used by HLM. The authors are aware of the problem and will correct it in the next release.

Looking at the different solutions given by HLM in Table 3, the most visible changes are that of the intercept, from negative -4.53 to positive +4.06, and that of the correlation between the intercept (base) and the slope of IQ, from negative -.9846 to positive +.4168. A lower correlation often means a faster convergence and a more precise estimate of the parameters. For the same reason the VARCL starts its calculations with deviation scores, but recalculates the estimates back to the original values for raw data scores after convergence. Since HLM starts with the raw data, the user is sometimes forced to center the scores to obtain a solution. But the reason for using deviation scores instead of raw scores in our case was related to the different range of the variables, since IQ has a range of 60 to 130, while the other two variables (SEX and SES) have a much smaller range. To get a solution for our GALO example, it was sufficient to make the range of the variable IQ smaller and thus more equal to the scale of the other variables (SEX with two and SES with six integer categories).

Table 3: GALO data, HLM results with different scalings of IQ

	IQ/10	IQ/100	IQDev	IndVarDev	AllVarDev
FixedPart					
(gamma)00	-4.5390	-4.5420	3.5190	4.0660	-.0682
(gamma)10	-.0604	-.0599	-.0611	-.0604	-.0603
(gamma)20	.7830	7.831	.0783	.0783	.0783
(gamma)30	.1496	.1497	.1493	.1496	.1496
RandomPart					
(sigma)	.8786	.8782	.8783	.8785	.8785
(omega)00	.7551	.7676	.1859	.1328	.0532
(omega)11	.0411	.0424	.0421	.0411	.0415
(omega)22	.0155	1.559	.0002	.0002	.0002
(omega)33	.0012	.0011	.0010	.0012	.0012
Additional					
TotIter*	101	99	116	103	98
Groups	36	37	36	36	36
Correlation**	-.9846	-.9833	-.9355	.4191	.4168
Deviance	3583.20	3578.60	3587.79	3587.81	3587.81

* Stopping criterion is 0.00005.
** Corr is correlation between St(BASE) and S2(IQ).

The results of the different manipulations are shown in Table 3. The first column of this table shows a solution with IQ divided by 10, the second with IQ divided by 100, the third with IQ in deviation from the grand mean (which makes the range of the IQ-variable much smaller), and the fourth with all regressors in deviation from the grand mean. In the last column all variables are put in deviation of their respective grand mean (as we did in Table 2a), and as a result, the intercept is close to zero. In the first two columns of Table 3 the interpretation of the intercept is equal; it is the value for the dependent variable (advice) when the values of the independent variables (SES and SEX) are zero. Since SES and IQ are never zero, this situation does not exist. The interpretation of the intercept in column three is diffferent: When a student has a zero SEX (male) score, a mean IQ, and a zero SES (an unrealistic assumption), the predicted advice will be 3.5190 (the value of the intercept). The fourth column predicts a value of 4.0660 for advice for a student scoring average on all three independent variables. The changing values tor the slope of IQ in Table 3 over solutions, (gamma)20 and (omega)22, are due to the different scalings of IQ. The changes in the variance of the intercept, omega(00), and the changes of the correlation in the same table can easily be proven to be the result of using deviation scores instead of raw scores. These changes do not cause the conclusions based on one of these solutions to differ.

Researchers may prefer centering in some form because of the ease of interpretation. Some other reasons for centering (around the group mean in that case) are discussed by Raudenbush (1989a,b), Longford (1989b), and Plewis (1989). We do not enter this discussion, but only show the differences in interpretation of the intercept.

We also illustrate one tricky effect of speed comparisons between machines. Let us first determine what VARCL does with GALO data on the Sun 3/280 (using Sun's FTP floating-point accelerator, compiled with the standard f77 compiler). The programs were run using a 1200 baud serial connection to a Mac II, at a time when there was not much activity on the system. VARCL took 14 seconds between the definition of the maximal model and the output of the sample means, which is the stage in which the data are read and the sufficient statistics are computed. This time included waiting time for one user response and sending some output through the serial line. Computing initial estimates took 6 seconds, and the four iterations of the random intercept model took 11 seconds. For the slightly more complicated model in which the variances of the slopes and the covariances of the slopes with the intercept were nonzero, we needed 8 iterations and 36 seconds.

In a second run on the Sun 3/280, at about the same time of day, we first made a file that contained answers to all the questions asked by VARCL and made the program take its input from the file. Moreover, all output was also written to a file. This meant that no human reactions and no slow serial lines were involved; all input and output was done locally (time reading from and writing to disk were still included, of course). The improvements in time were dramatic. Input took 2 seconds, computing initial estimates only took one second, and all 4 iterations for the random intercept model were completed within the same second. For the more complicated random slope model, the 8 iterations took only 3 seconds.

3.2.3 GRAY

These are data on 864 students in 16 inner city schools in London (see Aitkin & Longford, 1986). We used two individual level variables, gender (SEX) and test score (VRQ). Again there was no school level variable, and we used random slope and a random intercept model.

The within group-model was

$$(\underline{ILEA})_i = \underline{\beta}_{0j} + \underline{\beta}_{1j}(SEX)_i + \underline{\beta}_{2j}(VRQ)_i + \underline{\varepsilon}_i,$$

and the between-group model was

$$\underline{\beta}_{0j} = \gamma_{00} + \underline{\delta}_{0j},$$
$$\underline{\beta}_{1j} = \gamma_{10} + \underline{\delta}_{1j},$$
$$\underline{\beta}_{2j} = \gamma_{20} + \underline{\delta}_{2j}.$$

Table 4 gives results for the random intercept and random slope models. Since HLM did not run on the raw data again, we used data in which all variables were centered around the grand mean. The intercept was close to zero as a result.

Table 4a: GRAY data, random intercept model

	GENMOD	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	-60.7358	-0.0274	-60.7400	-60.7700	-60.7690
(gamma)10	0.9079	0.9080	0.9079	0.9102	0.9101
(gamma)20	0.8243	0.8243	0.8243	0.8246	0.8246
RandomPart					
(sigma)	94.0966	94.0991	94.1000	93.8900	93.8852
(omega)00	2.0370	2.0281	2.0370	1.7790	1.7790
Additional					
TotalIter	27	4	5	4	5
TimeIter	3	1	17	10	2
deviance		6396.99			6387.44

* For HLM all variables were centered around their grand means.

In Table 4b we used another option in order to get HLM running: We divided the IQ variable (VRQ) by 100. As a result, the estimate of the slope gamma(20) was 100 times larger and the estimate of (omega)22 was 10000 times larger than it was for the other programs. Taking this into account, we see that the solutions of the five programs are very close, with the main difference found between the RML and FML programs and, for omega(11), between the two ML2 programs and the rest. Again, the largest difference between packages was in speed and number of iterations. A slightly different version of these data, with 907 students in 18 schools, was also run with VARCL on the Sun 3/280 and the Vax 11/750. The random intercept model took 5 iterations and 18 seconds (Sun) or 25 seconds (Vax); the random slope model took 8 iterations and 29 seconds (Sun) or 42 seconds (Vax). Both Sun and Vax were measured through a 9600 baud direct connect line to a Mac II. Solutions were the same as in Table 4.

Table 4b: GRAY data, random slope model, no macro variable

	GENMOD	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	-61.2623	-61.2634	-61.3700	-61.2700	-61.4642
(gamma)10	0.7939	0.7936	0.7755	0.7857	0.7703
(gamma)20	0.8305	83.0510*	0.8319	0.8309	0.8328
RandomPart					
(sigma)	91.4333	91.4273	91.5400	91.4800	91.3461
(omega)00	160.3000	160.3386	144.8000	124.5000	143.8160
(omega)11	0.5124	0.5432	0.0000	0.0000	0.4740
(omega)22	0.0176	176.5044*	0.0183	0.0160	0.0170
Additional					
TotalIter	647	301	8	6	12
TimeIter	9	3	81	63	3.3
deviance		6373.98			6374.77

* Coefficients are based upon VRQ/100.

3.2.4 OSHEA

The data in this section were provided by David O'Shea, Graduate School of Education, UCLA. Individuals were 4313 UCLA graduates working in 12 different industries (elementary or secondary school; college / university / technical institute / professional school; retail / wholesale; human services organization; local government; other business or service establishments; commerce / insurance / finance / real estate; U.S. military service; agriculture / mining; US government / civilian employee; manufacturing / construction; transportation / public utilities). Variables on the individual level were gender (SEX), parental income (PI), selectivity of the major (SEL), education (EDU), GPA, occupation (OCC), incentive (INC), hours/day (HD), Q14B, and human capital (HC). The dependent variable was income. There were no variables on the group level. The example is interesting, because there were many predictors and large groups.

The within group-model was

$$\underline{(\mathrm{INCOME})}_i = \beta_{0j} + \beta_{1j}(\mathrm{SEX})_i + \beta_{2j}(\mathrm{PI})_i + \beta_{3j}(\mathrm{SEL})_i + \beta_{4j}(\mathrm{EDU})_i + \beta_{5j}(\mathrm{GPA})_i + $$
$$+ \beta_{6j}(\mathrm{OCC})_i + \beta_{7j}(\mathrm{INC})_i + \beta_{8j}(\mathrm{HD})_i + \beta_{9j}(\mathrm{Q14B})_i + \beta_{10,j}(\mathrm{HC})_i + \varepsilon_i,$$

and the between-group model allowed for two random coefficients

$$\beta_{0j} = \gamma_{00} + \delta_{0j},$$
$$\beta_{1j} = \gamma_{10} + \delta_{1j},$$

The other coefficients were fixed (i.e., $\beta_{s,j} = \gamma_{s0}$ for s=2,...,10).

Table 5: OSHEA data, random intercept model

	GENMOD	HLM*	ML2** (R)	ML2** (F)	VARCL
FixedPart					
(gamma) 00	-2.4171				-2.4134
(gamma) 10	-1.4465				-1.4486
(gamma) 20	0.0200				0.0200
(gamma) 30	0.2324				0.2326
(gamma) 40	1.0453				1.0446
(gamma) 50	0.2656				0.2655
(gamma) 60	0.6132				0.6128
(gamma) 70	0.8990				0.9007
(gamma) 80	0.4070				0.4073
(gamma) 90	0.7936				0.7933
(gamma) 10,0	0.2758				0.2757
RandomPart					
(sigma)	11.4931				11.4689
(omega) 00	6.0550				5.4370
(omega) 11	1.1400				1.0220
Additional					
TotalIter	6				7
Time iter	7.5				4.2

* For HLM we got a runtime error (M6101: Math floating point error: invalid centering)
** ML2 gave the error message Worksheet Full

Only GENMOD and VARCL reached a solution for this data set. We do not precisely know why. The problem was caused by either the combination of a large sample and a large number of variables, or the large number of observations within a group. The model that was fitted was fairly simple: only one random slope and no second level interaction variables. At the same time, the number of observations within groups was large, which made the number of iterations needed to reach the stop criterion small.

3.2.5 VOCAL

The individuals in this data set were 5310 students in 70 secondary schools in the city of Amsterdam in 1975. Individual level predictors were SEX and CITO. (CITO is a school attainment test taken at the beginning of secondary education.) The dependent variable was CAREER, which was a scale based on a multiple correspondence analysis of the complete school career. For details we refer to Kreft (1987). A school-level predictor was TYP (which takes five different values, one for each of the five major types of secondary education in The Netherlands). Although this is not entirely appropriate, TYP was not used as a categorical variable, but as a continuous variable.

The model we used in this study was

$$\underline{(\text{CAREER})}_i = \underline{\beta}_{0j} + \underline{\beta}_{1j}(\text{SEX})_i + \underline{\beta}_{2j}(\text{CITO})_i + \underline{\varepsilon}_i,$$

and the full between-group model was

$$\underline{\beta}_{0j} = \gamma_{00} + \gamma_{01}(\text{TYP})_j + \underline{\delta}_{0j},$$
$$\underline{\beta}_{1j} = \gamma_{10} + \gamma_{11}(\text{TYP})_j + \underline{\delta}_{1j},$$
$$\underline{\beta}_{2j} = \gamma_{20} + \gamma_{21}(\text{TYP})_j + \underline{\delta}_{2j}.$$

As usual, we fit a random intercept version,

$$\underline{\beta}_{0j} = \gamma_{00} + \gamma_{01}(\text{TYP})_j + \underline{\delta}_{0j},$$
$$\underline{\beta}_{1j} = \gamma_{10} + \underline{\delta}_{1j},$$
$$\underline{\beta}_{2j} = \gamma_{20} + \underline{\delta}_{2j},$$

and a random slope version,

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(TYP)_j + \delta_{0j},$$
$$\beta_{1j} = \gamma_{10},$$
$$\beta_{2j} = \gamma_{20} + \gamma_{21}(TYP)_j + \delta_{2j}.$$

Results are given in Tables 6a and 6b.

Table 6a: VOCAL data, random intercept model

	GENMOD	HLM*	ML2(R)	ML2(NR)	VARCL
FixedPart					
(gamma)00	1.2360	1.2360	1.2360	1.2340	1.2326
(gamma)01	-.2742	-.2743	-.2742	-.2731	-.2727
(gamma)10	-.0798	-.0797	-.0797	-.0797	-.0796
(gamma)20	-.3229	-.3229	-.3229	-.3232	-.3231
RandomPart					
(sigma)	0.3995	0.3995	0.3995	0.3995	0.3995
(omega)00	0.3484	0.3480	0.3483	0.3392	0.3563
(omega)10	0.0071	0.0070	0.0071	0.0068	0.0070
(omega)20	0.0962	0.0962	0.0962	0.0946	0.0960
Additional					
TotalIter	109	101	17	12	11
TimeIter	145	13	567	413	12
Deviance		10628			10593

It is clear from the Tables 6a and 6b that the packages gave very similar results, for fixed as well as random parts, for the first level as well as for the interaction coefficients. The main difference was again between restricted and full and between VARCL and the rest, but the differences were small. We expected that much, since the data were well conditioned and a large number of observations within and between were present. The programs did not behave as nicely in the following data set, which was substantially smaller.

Table 6b: VOCAL analysis with TYPE as macro variable

	GENMOD	HLM	ML2 (R)	ML2 (NR)	VARCL
FixedPart					
(gamma) 00	1.5041	1.5040	1.5040	1.5020	1.5026
(gamma) 10	-.0981	-.0981	-.0981	-.0981	-.0981
(gamma) 01	-.3728	-.3728	-.3723	-.3723	-.3722
(gamma) 20	-.1221	-.1221	-.1221	-.1221	-.1219
(gamma) 21	-.8340	-.8340	-.8350	-.8350	-.8360
RandomPart					
(sigma)	0.4016	0.4C16	0.4016	0.4015	0.5016
(omega) 00	0.3591	0.3590	0.3590	0.3480	0.1650
(omega) 20	0.0899	0.0899	0.0899	0.0869	0.0870
Additional					
TotalIter	8	11	6	6	14
TimeIter	81	10.2	454	256	8.8
Deviance		10632			10599

3.2.6 WEBB

This set comprised data from 96 students (grades 7 and 8) in three average-ability Los Angeles junior high schools. They were in 35 small groups. The example (data provided by Noreen Webb, Graduate School of Education, UCLA) is interesting, because the number of groups was relatively large, and thus the number of individuals per group was small. Individual level variables were posttest (POST), which is the dependent variable, pretest (PRE), and an interaction variable (NOA: asking a question and not getting an answer). For further details and discussion we refer to Webb (1982).

The group level variable was the pretest means in the group (PREM). The model was

$\underline{(POST)}_i = \beta_{0j} + \beta_{1j}(PRE)_i + \beta_{2j}(NOA)_i + \varepsilon_i$, and the between-group model was

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(\text{PREM})_j + \delta_{0j},$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}(\text{PREM})_j + \delta_{1j},$$
$$\beta_{2j} = \gamma_{20} + \gamma_{21}(\text{PREM})_j + \delta_{2j}.$$

After some preliminary exploration we decided on a model with both $\delta_{1j} \equiv 0$ (coefficient of PRE is non-random) and $\gamma_{11} = 0$ (no effect of pretest mean on pretest slope). Thus the single-equation specification of the model was

$$\underline{(\text{POST})}_i = \gamma_{00} + \gamma_{01}(\text{PREM})_j + \gamma_{10}(\text{PRE})_i + \gamma_{20}(\text{NOA})_i + \gamma_{21}((\text{NOA})_i(\text{PREM})_j) +$$
$$+ \{\delta_{2j}(\text{NOA})_i + + \delta_{0j} + \varepsilon_i\}.$$

Results are given in Table 7b. Table 7a is the random slope model, which has in addition $\gamma_{21} = 0$ (no effect of pretest mean on NOA slope).

Table 7a: WEBB data, random slope model, no macro variables

	GENMOD***	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00		19.6895	20.5900	20.5500	20.3772
(gamma)01		-1.2115	-1.2050	-1.2220	-1.1761
(gamma)10		3.3030	3.2150	3.2220	3.2191
(gamma)20		-3.9235	-4.1170	-4.0640	-4.1021
RandomPart					
(sigma)		26.2841	26.2400	25.6100	25.8873
(omega)00		40.8725	44.5400	42.7800	2925.1110**
(omega)20		4.0382	4.6570	4.4000	3.7100
Additional					
TotalIter		200	17	26	13
TimeIter		4.3	8.2	5.8	4.3
deviance		620.2			612.0

* Stopped at 200 iterations.
** Obviously a bug.
*** GENMOD does not work because of singularities

In this example of group interaction within school classes, the number of observations per group was very small. The number of variables outnumbered the number of observations per group. Traditional models that use LS estimation cannot estimate parameters within groups . GENMOD did not run either and gave an error message about singular matrices.

Table 7b: WEBB data, random slope model, PREM macro variable

	GENMOD***	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00		10.9277	11.6600	11.3500	12.0932
(gamma)01		0.2545	0.0764	0.0921	0.0171
(gamma)10		3.2786	3.3420	3.3540	3.3428
(gamma)20		0.1677	0.1204	0.2575	-0.0959
(gamma)21		-0.6824	-0.6723	-0.6823	-0.6423
RandomPart					
(sigma)		25.8800	26.1700	26.3300	25.7877
(omega)00		43.4384	46.1800	43.8600	3000.8270**
(omega)20		4.4773	4.8680	4.4580	3.8280
Additional					
TotalIter		200	26	168	13
TimeIter		5.3	3.5	6.4	4.7
deviance		619.9			611.2

* Stopped at 200 iterations.
** Obviously a bug.
*** GENMOD does not work because of singularities

3.3 Conclusions.

It is difficult to summarize the results of our numerical comparisons, but we shall try to give the main conclusions. Some of them are trivial, such as the conclusion that some computers are faster than others and the conclusion that writing directly to the screen or a disk file is faster than sending data over telephone lines. But there are a number of conclusions that appear to be quite useful.

EM algorithms for (co)variance component analysis are helpful, because they are relatively simple to program (especially in array oriented interpreted languages such as APL, MATHLAB, GAUSS), because they give monotone convergence, and because they always stay within the boundaries of the parameter space. There convergence can be tediously slow, especially for more complicated models. HLM uses Aitkin acceleration, which makes quite a 'arge difference, and we believe that even better acceleration algorithms are possible (Jamshidian & Jennrich, 1990). The convergence of GENMOD is sometimes intolerably slow. We are investigating the possibility of a bug that causes this, although it is due in part to the very strict convergence criteria in GENMOD. The scoring algorithm of VARCL will tend to give much faster convergence, although sometimes various parameters of the process have to be adjusted because of singularity, boundary conditions, negative eigenvalues, and divergence. We have observed sublinear convergence of VARCL in various examples, probably a result of excessive damping of the upgrade. Using the results of Lindstrom and Bates (1988) could very well produce a more robust implementation of the scoring algorithm. ML 2 has an algorithm that is still somewhat of a mystery to us, but it works quite well in almost all cases. Because all the data have to be kept in core, it cannot analyze really large examples. We think that this a high price to pay for the relatively small gain in additional generality.

In general, it follows from our analysis that two-level models with random slopes have very complicated likelihood surfaces. Maximizing the likelihood is inherently a difficult problem unless the model is approximately true and sample size is really large (in that case, OLS will give very good starting values). We also think that there is much room for exploring alternatives to ML such as the weighted least squares methods discussed by De Leeuw and Kreft (1986), which are possibly somewhat more robust, or MINQUE/MINVE methods (Rao & Kleffe, 1989). Investigators (if the past is any indication) will tend to choose models that are too complicated (five levels, with 10 variables on each level). This leads to impossibly difficult search problems over the space of models and to impossibly difficult likelihood maximization problems. None of the programs reviewed here can handle such problems gracefully—something would be wrong if they could.

We have found a variety of bugs in the programs. Some of them have been corrected, some of them remain (such as the value of the likelihood function in GENMOD, or the test for singularity in HLM), but none are very serious. All four programs tend to converge to the same solutions, which is rather nice, although there are some unpleasant exceptions.

In our comparisons we have not addressed the question about the usefulness of the statistical information: Are the likelihood ratios close to chi-squares? How accurate are the standard errors? Do the estimates really improve the mean square error of OLS and WLS estimates? Such questions are important, in fact more important than computational speed or a friendly interface, but they require more complicated research. Once you know that hierarchies exist, you see them everywhere. Thus the applicability of the software seems almost unlimited. This pleases the authors of the programs, who have no interest in pointing out limitations and shortcomings of their products. We think that it is time to start sampling, resampling, and cross validation studies to get a more realistic idea about the possibilities of the techniques.

4. MULTIPATH.

The comparisons in this report, together with the requirements of the "evaluation of primary education in The Netherlands," indicate that from the theoretical and practical point of view, a number of developments in multilevel analysis are very desirable. We shall therefore put them on the agenda of our project. Some of our concerns have to do with improving and studying existing software; some of them address the development of a new software program, MULTIPATH.

The existing programs for multilevel analysis reviewed in this report must be compared in various ways; a numerical comparison is particulary needed. There are some unresolved questions we have not answered, and we suspect the presence of bugs, which seem to affect the programs when they are run through different compilers. As far as VARCL and GENMOD are concerned, the code of these programs is in the public domain, and we shall have ample opportunity to review these programs in more detail. The situation for HLM is a bit more complicated because the program and manual are now published by a software house and the code is no longer in the public domain. For ML2 we do not have any code, and we can only let the authors know which enhancements we would like to see.

In regard to the loss function, we have seen that although all four packages use ML, they differ in to what they apply the ML principle (either to the raw data, as in FML, or to the least squares residuals, as in RML). This could be taken a bit further, perhaps, especially in models with more than one level, because there are many ways in which residuals can be defined. Moreover, we have seen that the likelihood function can be expressed in various ways, and we shall look further into comparison of the resulting formulas (both from the computational and the interpretational point of view). We also emphasize here that among statisticians, ML methods for variance component analysis are not necessarily thought to be optimal. The books by Humak (1984) and by Rao and Kleffe (1989) concentrate on MINQUE and MINVE and related methods, methods that have exact small sample optimality properties.

As far as algorithms are concerned, it seems that our (admittedly preliminary) results so far suggest that EM algorithms have limited usefulness, unless a suitable accelerator is provided. Accelerating EM is an active area of research (Meilijson, 1989, Jamshidian and Jennrich, 1990), and we hope to experiment somewhat using GENMOD. Newton-Raphson[1] and scoring algorithms seem to work quite well, but they have the disadvantage that they are general purpose algorithms that do not take the structure of the problem into account. For this reason, for instance, they can produce variance estimates of less than zero (although it is not difficult to prevent this by suitable parametrization). In that sense both EM and the iterative weighted alternating LS methods of ML2 are better.

We need more generality for the project we are dealing with. In particular, we need more levels, and we need a more general class of models. This will be incorporated in the design of the program MULTIPATH. Because of obvious interpretational difficulties, we think that it is unwise to go beyond three levels; however, in principle, software can be written that handles an unlimited number of levels. For the class of models we shall restrict ourselves in MULTIPATH to recursive path models with observed variables. It is not too difficult to make multilevel versions of general structural equations models (in fact we already have the theory for this), but we think that this involves a step that is too big. General structural equation techniques already have their share of problems, and compounding these by overlaying a multilevel structure seems too risky.

All four programs have command-line interfaces and run on XT and/or AT types of computers. It is not too difficult (although rather time-consuming) to replace these interfaces by character-based menu-driven or even graphical interfaces, using existing libraries for various machines. To maximize portability we will write the program MULTIPATH in the C language, and we will make a portable command line version (which will run on PC, PS, Mac, VM/CMS mainframes, and Unix boxes). If there is enough time and money, we shall try to build versions for X-Windows, MS-Windows, OS2, and Mac Toolbox, using a portable graphical interface library such as XVT.

[1] Raphson was Newton's computer programmer.

5. REFERENCES

Aitkin, M.A., & Longford, N. (1986). Statistical modelling issues in school effectiveness studies. *Journal of the Royal Statistical Society, 149A*, 1-43.

Bock, R.D. (Ed.). (1988). *Multilevel analysis of educational data.* San Diego: Academic Press.

Boyd, L.H., & Iversen, G.R. (1979). *Contextual analysis: Concepts and statistical techniques.* Belmont: Wadsworth.

Bryk, A.S., & Raudenbush, S.W. (1987). Applying the hierarchical linear model to measurement of change problems. *Psychological Bulletin, 101*, 147-158.

Bryk, A.S., Raudenbush, S.W., Seltzer, M., & Congdon, R.T. (1988). *An introduction to HLM: Computer program and users' guide.* Chicago: University of Chicago.

Burstein, L. (1980). The analysis of multilevel data in educational research and evaluation. In D. Berliner (Ed.), *Review of research in education, volume 8.* Washington, DC: American Educational Research Association.

Burstein, L., Kim, K-S., & Delandshere, G. (1988). *Multilevel investigations of systematically varying slopes: Issues, alternatives, and consequences.* In R.D. Bock (Ed.), *Multilevel analysis of educational data.* San Diego: Academic Press

Cliff, N. (1987). *Analyzing multivariate data.* Orlando: Harbourt Brace Jovanovich.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B39*, 1-38.

Dempster, A.P., Rubin, D.B., & Tsutakawa, R.K (1981). Estimation in covariance components models. *Journai of the American Statistical Association, 76*, 341-353.

Everett, B.S., & Dunn, G. (1983). *Advanced methods of data exploration and modelling.* London: Heinemann Educational Books Ltd.

De Leeuw, J., & Kreft, I. (1986). Random coefficient models for multilevel analysis. *Journal of Educational Statistics, 11*, 57-85.

Goldstein, H. (1986). Multilevel mixed linear model analysis using iterative generalized least squares. *Biometrika, 73*, 43-56.

Goldstein, H. (1987). *Multilevel models in educational and social research.* London: Griffin.

Goldstein, H. (1989). Restricted unbiased iterative generalized least squares estimation. *Biometrika, 76*, 622-623.

Harville, D.A. (1974). Bayesian analysis for variance components using only error contrasts. *Biometrika, 61*, 383-385.

Harville, D.A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association, 72*, 320-340.

Healy, M. (1987). *Nanostat users' guide.* Unpublished documentation.

Hemelrijk, J. (1965). Underlining random variables. *Statistica Neerlandica, 20*, 1-8.

Horng, S.C. (1987). Examples of sublinear convergence of the EM algorithm. *Proceedings of the Statistical Computing Section, ASA*, 266-271.

Hox, J.J., Kreft, I., & Hermkens, P.L.J. (1989). *Factorial surveys: An example of multilevel design*. Amsterdam: Methodenleer en statistiek, MLS-Publicatie 40.

Humak, K.M.S. (1984). *Statistische Methoden der Modellbildung* (Vol. III). Berlin: Akademie Verlag.

Jamshidian, M., & Jennrich, R.I. (1990). *Conjugate gradient acceleration of the EM algorithm* (UCLA Statistics Series, No. 48). Los Angeles: University of California, Los Angeles.

Jennrich, R.I., & Schluchter, M.D. (1986). Unbalanced repeated-measures models with structured covariance matrices. *Biometrics, 42*, 805-820.

Kim, K-S., & Kreft, I. (1989). *A version of HLM written in GAUSS* (Internal publication). Los Angeles: UCLA Graduate School of Education.

Kreft, I. (1987). *Methods and models for the measurement of school effects*. Dissertation, University of Amsterdam.

Kreft, I., & De Leeuw, E.D. (1988). The See-Saw effect: A multilevel problem? *Quality and Quantity, 22*, 127-137.

Kreft, I., & De Leeuw, J. (1989). Model based rankings of schools (UCLA Statistic Series, No. 7). *International Journal of Education,* in press.

Laird, N.M., & Ware, J.H. (1982). Random effects models for longitudinal data. *Biometrics, 38*, 963-974.

LaMotte, L.R. (1972). Notes on the covariance of a random, nested ANOVA model. *Annals of Mathematical Statistics, 43*, 659-662.

Lindstrom, M.J., & Bates, D.M. (1988). Newton-Raphson and EM algorithms for linear mixed-effects models for replicated measures data. *Journal of the American Statistical Association, 83*, 1014-1022.

Lockheed, M.E., & Longford, N.T. (1989). *A multilevel model of school effectiveness in a developing country*. Unpublished manuscript, The International Bank for Reconstruction and Development/The World Bank.

Longford, N.T. (1987). A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. *Biometrika, 74*, 817-827.

Longford, N.T. (1988, February). *A quasilikelihood adaption for variance component analysis.* Princeton, NJ: Educational Testing Service.

Longford, N.T. (1986). *Statistical modelling of data from hierarchical structures using variance component analysis*. Program developed at the Center for Applied Statistics, Lancaster University.

Longford, N.T. (1988). Fisher scoring algorithm for variance component analysis of data with multilevel structure. In R.D. Bock (Ed.), *Multilevel analysis of educational data* (p. 103). San Diego: Academic Press

Longford, N.T. (1989a). Contextual effects and group-means. *Multilevel Modelling Newsletter, 1*(3), 5.

Longford, N.T. (1989b). To center or not to center. *Multilevel Modelling Newsletter, 1*(3), 7.

Mason, W.M., Wong, G.Y., & Entwisle, B. (1984). Contextual analysis through the multilevel linear model. *Sociological Methodology*, 72-103.

Mason, W.M., Anderson, A.F., & Hayat, N. (1988). *Manual for GENMOD*. Ann Arbor: Population Studies Center, University of Michigan.

Meilijson, I. (1989). A fast improvement to the EM algorithm on its own terms. *Journal of the Royal Statistical Society, B51*, 127-138.

Muthen, B.O. (1989). Latent variable modeling in heterogeneous populations. *Psychometrika*, in press.

Muthen, B.O. & Satorra, A. (1988). Multilevel aspects of varying parameters in structural models. In R.D. Bock (Ed.), *Multilevel analysis of educational data* (p. 103). San Diego: Academic Press

Oberhofer, W., & Kmenta, J. (1974). A general procedure for obtaining maximum likelihood estimates in generalized regression models. *Econometrica, 42*, 579-590.

Plewis, I. (1989). Comment on "centering" predictors in multilevel analysis. *Multilevel Modelling Newsletter, 1*(3), 6.

Rao, C.R., & Kleffe, J. (1989). *Estimation of variance components*. Amsterdam: North Holland.

Rabash, J., Prosser R., & Goldstein, H. (1989). *ML2: Software for two-level analysis users' guide*. London: Institute of Education, University of London.

Raudenbush, S.W. (1987). Educational applications of hierarchical linear models: A review. *Journal of Educational Statistics, 13*, 85-116.

Raudenbush, S.W. (1989a). "Centering" predictors in multilevel analysis: Choices and consequences. *Multilevel Modelling Newsletter, 1*(2), 10-12.

Raudenbush, S.W. (1989b). A response to Longford and Plewis. *Multilevel Modelling Newsletter, 1*(3), 8-10.

Raudenbush S.W., & Bryk, A.S. (1985). Empirical Bayes metaanalysis. *Journal of Educational Statistics, 10*, 75-89.

Raudenbush S.W., & Bryk, A.S. (1986). A hierarchical model for studying school effects. *Sociology of Education, 59*, 1-17.

Raudenbush S.W., & Bryk, A.S. (1987). Examining correlates of diversity. *Journal of Educational Statistics, 12*, 241-269.

Schluchter, M.D. (1988). *BMDP5V: Unbalanced repeated measures models with structured covariance matrices* (Tech. Rep. No. 86). Los Angeles: BMDP Statistical Software.

Stram, D.O., Laird, N.M., & Ware, J.H. (1986). An algorithmic approach to the fitting of a general mixed ANOVA model appropriate in longitudinal settings. In *Computer science and statistics: Proceedings of the Seventeenth Symposium on the Interface.* Amsterdam: North Holland.

Van den Eeden, P., & Saris, W. (1984). Empirisch onderzoek naar multilevel uitspraken. *Mens en Maatschappij, 59*, 165-178.

Van der Leeden, R., & De Leeuw, J. (1990). *Comparison of four statistical packages for repeated measure and growth curve analysis.* Unpublished manuscript.

Webb, N.M. (1982). Group composition, group interaction and achievement in cooperative small groups. *Journal of Educational Psychology, 74*, 475-484.

Weisberg, S. (1985). *Applied linear regression.* New York: John Wiley and Sons.

Wong, G.Y., & Mason, W.M. (1985). The hierarchical logistic regression model for multilevel analysis. Extensions of the hierarchical normal linear model for multilevel analysis. *Journal of the American Statistical Association, 80*, 513-524.

Wong, G.Y., & Mason, W.M. (1989). *Ethnicity, comparative analysis and generalization of the Hierarchical Normal Linear Model for Multilevel Analysis* (Research Rep. No. 89-138). Ann Arbor: Population Studies Center, University of Michigan.