september 2002 (Free Software, for non-commercial purposes)
Version 1.2 *Manual Multilevel IRT Software.* In S-Plus 6 for Windows.
Available at: http://users.edte.utwente.nl/Fox
Jean-Paul Fox
Twente University
Faculty of Educational Science and Technology
PO. Box 217, 7500 AE Enschede
The Netherlands

Introduction

With this software the parameters of a multilevel IRT model can be estimated using binary and/or ordinal scored item responses. The multilevel model can handle manifest variables on different levels. For example, Level 1 consists of student characteristics and Level 2 of school characteristics. Item response models are used to relate the latent abilities to the test scores. Respondents are administered a survey to measure their abilities or attitudes to certain topics. Usually interest is focused on the correlation between the abilities underlying the tests and other manifest variables. Thus, the structural multilevel model is combined with one or more measurement models. Variables within the multilevel model may be latent but dichotomous or polytomous scored items are observed. All parameters of the multilevel model and the measurement model(s) are estimated simultaneously. Within this approach the standard errors of the parameter estimates are estimated correctly. Further, the information of the manifest variables reduces the estimation error of the person and item parameters.

The program consists of two files: *S.dll_170902* & *basicmlirt_170902.ssc*
Before one can use the program;

1. In “basicmlirt_170902” the position of the S.dll must be given, that is, fill in at line 430 and 450,
   dyn.open("c:\\…\\S.dll")
   dyn.close("c:\\…\\S.dll")
   (Double brackets, \\, are needed, which is common in S-Plus)
2. Open S-plus, to make the functions available within a S-plus session, type in the command window:
   source(file = "c:\\…\\basicmlirt_170902.ssc", echo = F)
   where now the position is needed of the file basicmlirt_170902.ssc.
3. Add the file *HPD.ssc* for the computation of confidence intervals
   source(file = "c:\\…\\HPD.ssc", echo = F)
4. Type in the command window *mlirtdialog2()* when obtaining the dialog for the first time. To obtain the dialog, type in the command window
   guiDisplayDialog("Function",Name="mlirt2").

The Program

The computations are programmed in Fortran 6 and formed in *S.dll*. S-Plus 6 is used to handle the input and the output of the S.dll. Splus is handy for importing large datafiles, for doing data-transformations, plotting results etc. A dialog has been made in S-Plus 6. Estimating the parameters of a multilevel IRT model is done by filling in the dialog. But there are some important remarks when filling in this dialog. The dialog is displayed in the following figure.

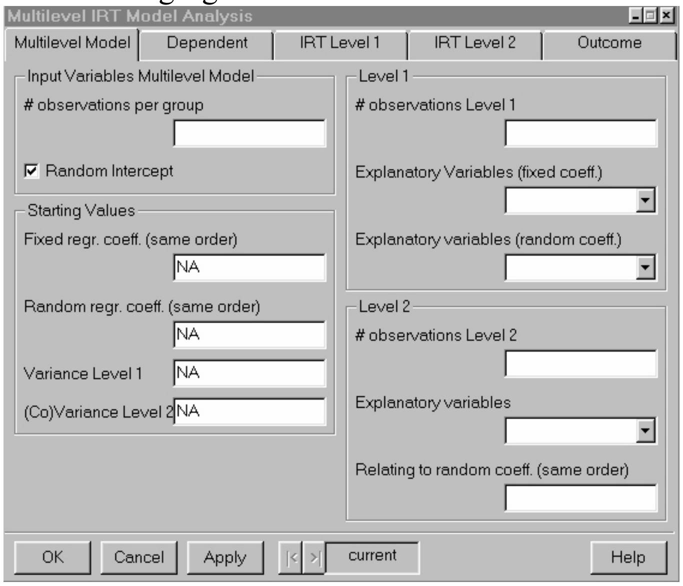

Figure 1: Page 1 multilevel IRT dialog

The data should be available in the working directory, and these data-objects should have the proper form, described below. The general form of the multilevel model is, with i=1,…,N Level 1 units and j=1,…,J level 2 units,

Equation 1

$$
\begin{aligned}
&\theta_{ij} = \beta_{0j} + \beta_{1j}X_{1ij} + \beta_{2j}X_{2ij} + \ldots + \beta_{Qj}X_{Qij} + e_{ij} \\
&\beta_{0j} = \gamma_{00} + \gamma_{01}W_{10j} + \gamma_{02}W_{20j} + \ldots + \gamma_{03}W_{30j} + u_{0j} \\
&\beta_{1j} = \gamma_{10} + \gamma_{11}W_{11j} + \gamma_{12}W_{21j} + \ldots + \gamma_{13}W_{31j} + u_{1j} \\
&\vdots \\
&\beta_{Qj} = \gamma_{Q0} + \gamma_{Q1}W_{1Qj} + \gamma_{Q2}W_{2Qj} + \ldots + \gamma_{Q3}W_{3Qj} + u_{Qj} \\
&e_{ij} \sim N\left(0, \sigma^2\right) \\
&u_j \sim N\left(0, T\right)
\end{aligned}
$$

The dialog consists of 5 pages, *Multilevel Model*, *Dependent, IRT Level 1, IRT Level 2, Outcome*. Each page will be described below, a corresponding figure will be given.

The first page displayed in Figure 1 defines the structural multilevel model. There are four blocks,

1. Input Variables Multilevel Model
   - #observations per group, Jx1 vector (J groups) in each element the number of observations
   - Random Intercept, true or not
2. Level 1
   - # observations Level 1, number or symbol, for example, N students.

- Explanatory variables (fixed coefficients), these are the <u>observed</u> explanatory variables at Level 1 with fixed coefficients. A vector (Nx1) Xf or if there are more than one explanatory variables, denote this as, c(Xf1,Xf2,…). In case of no observed explanatory variables fill in NA.
- Explanatory variables (random coefficients), these are the observed explanatory variables at Level 1 with fixed coefficients. A vector (Nx1) Xr or if there are more than one explanatory variables, denote this as, c(Xr1,Xr2,…). In case of no observed explanatory variables fill in NA.

3. Level 2
   - # observations Level 2, number or symbol, for example, J students.
   - Explanatory variables, these are the <u>observed</u> explanatory variables at Level 2 with fixed coefficients, otherwise a thirth Level is needed. A vector (Jx1) W or if there are more than one explanatory variables, denote this as, c(W1,W2,…). It is also possible to define the explanatory variables W as cbind(W1,W2,…) as one matrix of J times the number of variables in the command window. In case of no observed explanatory variables fill in NA.
   - Relating to random coefficient. The explanatory variables at Level 2 are predictors of a random coefficient at Level 1. A vector W as a predictor of the random intercept, corresponds with filling in 1. If it predicts β_{1j} this can be mentioned with a 2 etc. If a variable W is used as an explanatory variable for a random intercept and β_{1j} then fill in c(1,2) and state that two explanatory variables are used, although they are the same.
4. Starting Values
   - Fixed regression coefficients
   - Random regression coefficients, of length J
   - Variance Level 1, σ^2
   - Variance Level 2, Covariance matrix T.

Figure 2 presents the next page of the dialog, concerning the dependent variable of the structural model.

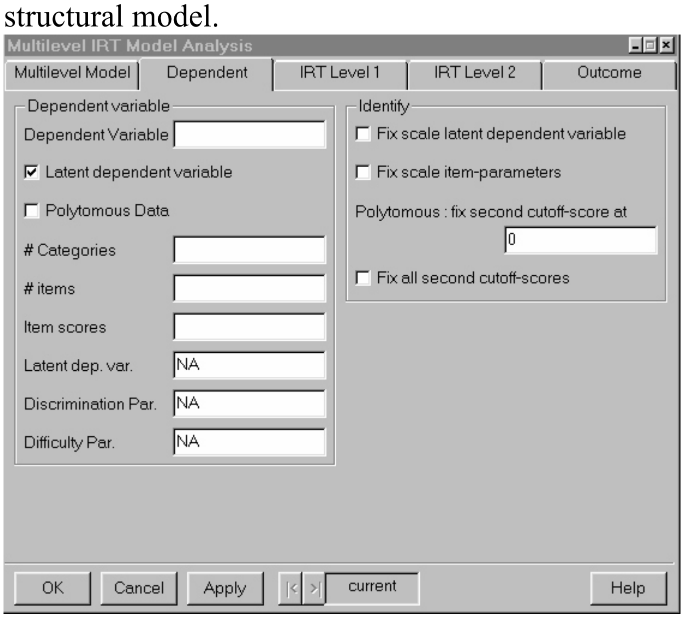

Figure 2: Page 2 multilevel IRT dialog

The block *Identify* gives three options how to identify the measurement model on the latent dependent variable. That is, the scale of the latent dependent variable is standard normal or the product of corresponding discrimination parameters equals one and the sum of discrimination parameters equals zero or the second cutoff score is fixed at some value, default zero. A unlucky choice of this value for the cutoff score can lead to numerical problems. It is also possible to fix all second cutoff scores. This can be helpful, for example, when unstable estimates are obtained using another identification method.

Within the block *Dependent Variable* the observed dependent variable can be given if it is not latent. If it is a latent variable or not this must be stated using the checkbox. If the dependent variable is latent, mark the next checkbox if polytomous data is used, give the number of categories and the name of the data-matrix comprehending the scored responses to a set of items. This matrix is assumed to be NxK, N persons responding to K items. Starting values for the latent variable, discrimination parameters and difficulty parameters can be given, to speed up the convergence of the algorithm. The default values are NA, meaning Non-Available.

The next page of the dialog describes the possibility of using latent explanatory variables at Level 1 using item response model(s) to measure the variables, see Figure 3.

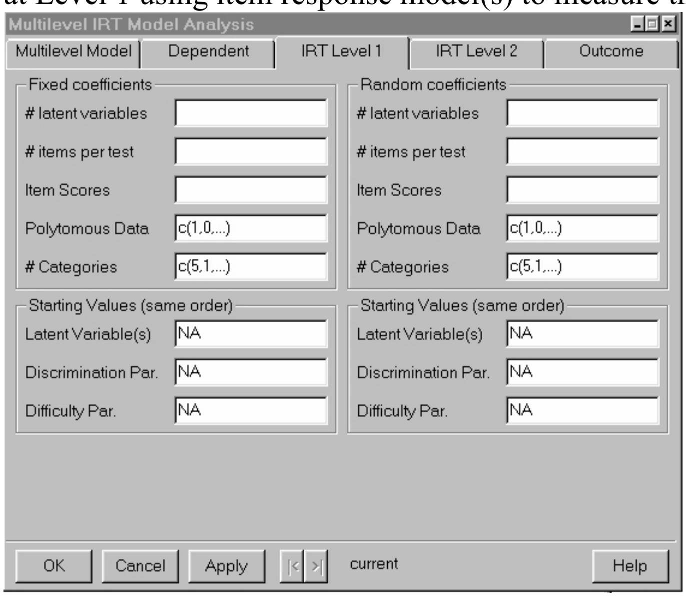

Figure 3: Page 3 multilevel IRT dialog

This page is split into two blocks, latent variable(s) with fixed coefficients (left) and latent variable(s) with random coefficients (right). For example, if a latent variable X at Level 1 has a random coefficent, fill in at the right,

- # latent variables   1
- # items per test   K, one test with K items
- Item Scores   X, say matrix X of dimension NxK, each row gives the result of a person responding to K items.
- Polytomous data   1 =True, 0=False, meaning dichotomous data.
- #Categories   4 if there are 4 categories, in case of dichotomous data fill in 1.

In case of two latent variables both with random coefficients,

- # latent variables   2
- # items per test   c(K1,K2), one test with K1 items and one with K2 items
- Item Scores   c(X1,X2), say matrix X1 of dimension NxK1, and X2 of NxK2.
- Polytomous data   1 =True, 0=False, meaning dichotomous data. Both dichotomous, c(0,0).

- #Categories c(1,1).

The order of giving the number of items per test, K1 & K2, and the variables with item scores, X1 & X2, must be the same, the program will link them. This resembles the way if the latent variable(s) had fixed coefficients. At the bottom starting values can be given for the latent variables and the item parameters, at the left, in case of fixed coefficents, at the right in case of random coefficents. Notice, if starting values are available for a specific latent variable, and there are more latent variables specified, starting values for all latent variables must be given in the same order, otherwise the program cannot link them.

Going further, it is possible to define latent variables at Level 2. Figure 4 shows the options to specify.

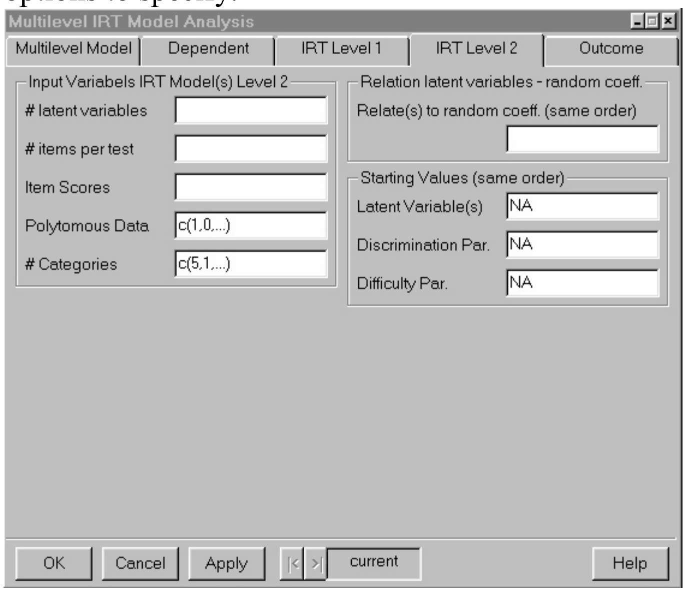

Figure 4: Page 4 multilevel IRT dialog

Again, specify the number of latent variables at Level 2, the number of items per correponding test, and the variables with the item scores, in the same order. Specify if the set of item scores contain polytomous or dichotomous data with a one or zero and specify the number of categories for each matrix with item scores, in the same order. Again starting values can be given. Each latent variable is an explanatory variable for a random coefficents, this is specified in the following way. A random intercept denotes 1, the next random coefficient is marked 2 etc. So, within the field denoted as *Relate(s) to random coefficient,* the defined latent explanatory variables are related to the random coefficents by stating their corresponding numbers, in the same order as they were defined. This method corresponds to how the observed explanatory variables at Level 2 were defined at Page 1 of the dialog.

At last, some specifications can be made concerning the MCMC algorithm. Figure 5 shows the options.

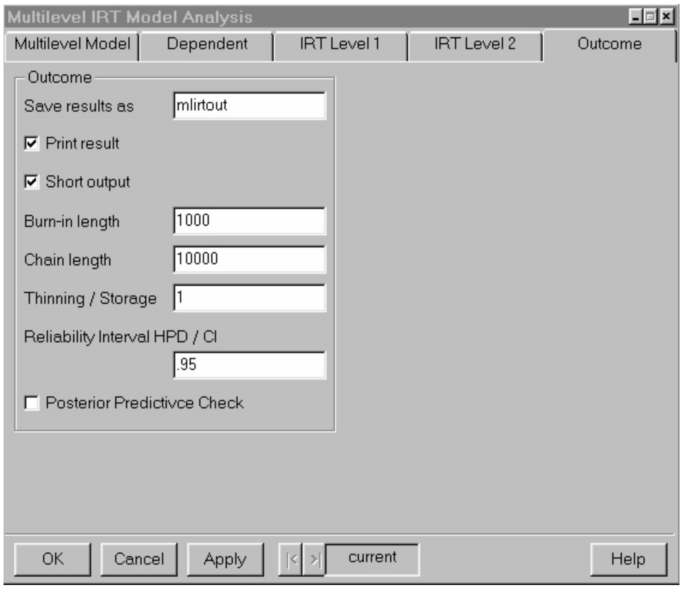

Figure 5: Page 5 multilevel IRT dialog

A name should be given, eventually, this contains the outcome of the MCMC algorithm. If marked, outcomes of the parameter estimates are printed in a report file, including standard deviations,highest posterior density intervals, and credibility intervals. A short output will not give the estimates and standard deviations of the item parameters. The burn-in length states how many of the first iterations should be ignored when estimating the parameters. The chain length defines the number of iterations. If thinning/storage equals one the sampled values in each iteration are stored, if it equals two, the values sampled in every second iteration are stored etc. Reliability intervals are computed according to a specified value. The parameter estimates are printed in a report file within S-Plus, the outcome in the specified object, default *mlirtout*, contains the sampled values. It is possible to perform some posterior predictive checks. They are fully described in "Multilevel IRT using Dichotomous and Polytomous Response Data". The output will give something like,

```
■  Statistics Based on Observed Data
       P-Value(s) Chi-Square Discrepancy      : 0.2023
       P-Value(s) Correlation Structure       : 0.1717
       Pseudo Bayes Factor(s)                 : -96773.0415  Total -96773.0415
       Log-Likelihood Multilevel Model        : -6386.8019.
```

The first statistic is the well-known χ^2-discrepancy to check if the replicated data under the model are close to the observed data. The second statistic can be used to test if the tetra or polychoric correlation in the predictive data are comparable to the correlation structure in the observed data. The second statistic returns the pseudo-Bayes factor corresponding to the observed data that can be used to compare (measurement) models. The log-likelihood of the structural multilevel model can be used to compare structural multilevel models with the same measurement models. So, the program can sample predictive data to do some model checking. Handling such an amount of data can be difficult within S-Plus. This can also be a problem when with a lot of IRT models all sampled item parameters are stored, thinning =1. For example, modeling responses of three tests each with 50 items, polytomously scored with 4 categories, corresponds to 3 IRT models, each with 50 discrimination parameters and 5*50 cuttoff parameters. This means that in each row 150 discrimination parameters are stored and in each row 5*50*3=750 cutoff parameters. In case of 10,000 iterations, this results in a matrix with 10,000 x 750 sampled cutoff values with 16 decimals. Caching such large matrices within S-Plus can be problematic. As a result, the posterior predictive checking is done in Fortran and only the above statistics are given as output. The program needs more time to do the computations.

The elements of the outcome variable, further on called as *mlirtout,* are defined in the appendix. It contains all sampled values. It is important to know how the values are stored, to obtain them in a proper way. This is explained in the appendix. In general, sampled values of variables with more than one element are stored one after another. That is, first all sampled values of the first element of the variable, then all sampled values of the second element etc. For example, S sampled values of a variable of dimension two, X=(X1,X2), are stored as follows, X11,X12,…,X1S,X21,X22,…X2S, where X11 is the first sampled value of X1. The S-Plus command matrix(X, ncol=2, nrow=S) stores all sampled values of X1 in column one and X2 in column 2.

Some remarks:

It is important that every input field gets a value, if it has no logical value then a 0 should be filled in. However, starting values and observed explanatory values have NA as the default value when they are missing. All other variables or specifications should get a 0 if no real value is present. The reason for this is that the dialog will give the input to S.dll and the input is matched by the total list of variables, empty cells are not passed through and this causes a mismatch with the total list of variables.

To get the output in a nice structured way, set the report font to *courier new 9 pt*.

Although a random coefficient should be defined at Level 1, it is not necessary to define any latent variables, this means that an ordinary multilevel model with two levels can be estimated.

The BUGS Output Analysis Program (BOA) software from http://www.public-health.uiowa.edu/boa can be used, within S-plus, for carrying out convergence diagnostics and statistical and graphical analysis of Monte Carlo sampling output.

By defining a .First function within the standard work directory of S-Plus with the statements to include both the multilevel IRT and the BOA software, both programs will be available every S-Plus session.

Example

An example is used to describe the output. The following example is described in Fox & Glas (2001). A dutch primary school mathematics test, 18 items, was taken by 2156 students distributed over 97 schools. 72 schools regularly participated in the school leaving examination, and are coded one the other 25 schools zero denoted in the variable

End. Three student characteristics were observed, socio-economic status (*SES*), non-verbal intelligence (*ISI*), and *gender* (boys=0,girls=1). The data are stored in the files:
math.dat : item scores, 18 columns (items), 2156 rows (persons).
predict1.dat : three Level 1 predictors, ISI, SES en Gender
predict2.dat : Level 2 indicator End, and the number of students per group.
The ascii data are read as follows,
Y <- matrix(scan(file=":\\…\\math.dat"),ncol=18,byrow=T)
X <- matrix(scan(file=":\\…\\predict1.dat"),ncol=3,byrow=T)
W <- matrix(scan(file=":\\…\\predict2.dat"),ncol=2,byrow=T)
Before filling in the dialog, notice that, K = 18 equals the number of items, N=2156 students, J=97 schools. Define W1<- W[,1] as the level 2 predictor and W2 <- W[,2] as the indicator for the number of students per school. The following outcome will arise, with 10,000 iterations and a burn-in length of 1,000, when estimating the multilevel IRT model ;

Equation 2

$$\theta_{ij} = \beta_{0j} + \beta_1 ISI + \beta_2 SES + \beta_3 Gender + e_{ij}$$
$$\beta_{0j} = \gamma_{00} + \gamma_{01} W_j + u_{0j}$$

Below the output will be explained in more detail. The model was identified by fixing the scale of the latent dependent variable to a standard normal scale.

```
Version 1.2 17/09/2002

**********                                                        **********
**********     Multilevel IRT Model                               **********
**********                                                        **********
**********     Wed Jun 04 15:12:43 WED 2003                       **********
**********                                                        **********
Theta_ij       =       B_0j    +3 B_f * X_f+ error_ij

B_ 0 j =G_ 0 0 +       G_ 0 1 *W_ 1    +              U_ 0

This can take some time       ...


**********                                                        **********
**********     Parameter Estimates Multilevel IRT Model           **********
**********                                                        **********
**********     Wed Jun 04 15:30:38 WED 2003                       **********
**********                                                        **********
```

The heading of the output contains the starting and stopping time, 10,000 iterations took about 18 minutes. Further a description of the model is given. Theta_ij denotes a latent dependent variable, otherwise it stated Y_ij, 3 B_f * X_f means three observed variables with fixed coefficients (ISI, SES, Gender). B_0j means a random intercept. W_1 denotes an explanatory variable at Level 2, in this case also called W.

```
              ***    Multilevel parameters  ***

-- Fixed Effects Level 1: --
        Mean(s)       : 0.4599 0.2512 -0.1845
        Sd(s)         : 0.0194 0.021 0.0384
        HPD           : [ 0.422 , 0.498 ]   [ 0.209 , 0.291 ]     [ -0.26 , -0.11 ]
```

```
        CI            : [ 0.422 , 0.498 ]    [ 0.21 , 0.292 ]       [ -0.26 , -0.11 ]
-- Fixed Effects Level 2 : --
        Mean(s)       : -0.3218 0.4876
        Sd(s)         : 0.0953 0.1074
        HPD           : [ -0.515 , -0.141 ]  [ 0.277 , 0.699 ]
        CI            : [ -0.51 , -0.133 ]  [ 0.277 , 0.699 ]
-- Variance Components, sigma: --
        Mean          : 0.4598
        Sd            : 0.0191
        HPD           : [ 0.422 , 0.497 ]
        CI            : [ 0.423 , 0.498 ]
-- Variance Components, T: --
        Mean(s)       : 0.173
        Sd(s)         : 0.0311
        HPD           : [ 0.115 , 0.235 ]
        CI            : [ 0.12 , 0.243 ]
```

The fixed effects at Level 1 are printed in the same order as defined, that is, according to equation 2, $\beta_1 = 0.460$ (ISI), $\beta_2 = 0.251$ (SES), and $\beta_3 = -0.185$ (Gender). The posterior standard deviations (Sd), 95% highest posterior density intervals (HPD) and 95% credible intervals (CI) are printed in the same order. Latent explanatory variables defined at Level 1 are printed after the coefficients of the observed variables and in the same order as the latent variables are defined.

The fixed effects at Level 2, first are the intercepts printed of all random coefficients, here, γ_{00}=-.3218, then the coefficients of the observed variables of the first random coefficient, here, γ_{01}=.4876, then the coefficients of the latent explanatory variables of the first random coefficient, then in the same order the coefficents of explanatory variables of the second random coefficent. The posterior sd's, HPD's and CI's are printed in the same order. The estimate of the variance coefficient at Level 1 follows directly. The estimated components of the covariance matrix are displayed by rows. Here, it follows that σ^2=.4598 and $\tau^2 = .173$.

Because the checkbox Posterior predictive check was marked, the output contains values of some statistics and p-values.

```
-- Statistics Based on Observed Data
        P-Value(s) Chi-Square Discrepancy    : 0.3178
        P-Value(s) Correlation Structure     : 0.6244
        Pseudo Bayes Factor(s)               : -19272.5927  Total -19272.5927
        Log-Likelihood Multilevel Model      : -3251.8842
```

Both p-values suggest that there is no concern that the model does not fit the data. The predicted data under the model resemble the observed data. The Pseudo Bayes factor and the log-likelihood value can be used to compare models.

Also, estimates of the item parameters are printed if no short output is specified. Both the discrimination and difficulty parameters are given and their posterior standard deviations.

```
               ***     IRT Model Parameters   ***

-- IRT Model Dependent Variable: --

******  Item parameters       ******
Discrimination parameters / Difficulty parameters
```

Item	mean	sd	mean	sd
1	0.83705	0.04598	-0.36751	0.03206
2	0.87473	0.05179	-0.89553	0.03978
3	0.75875	0.04095	-0.15810	0.02981
4	0.77546	0.04877	-0.99915	0.04054
5	0.48812	0.03630	-0.56934	0.03078
6	0.55159	0.03536	0.00036	0.02833
7	0.72537	0.04180	-0.47619	0.03186
8	0.65896	0.04891	-1.34940	0.04557
9	0.64851	0.03888	-0.31495	0.02976
10	0.59017	0.04083	-0.80765	0.03400
11	0.62622	0.04194	-0.84997	0.03602
12	0.47356	0.03477	-0.02664	0.02800
13	0.60529	0.05728	-1.63212	0.05640
14	1.04280	0.06171	-1.12725	0.04871
15	0.74561	0.04557	-0.73714	0.03517
16	0.54164	0.03819	-0.56448	0.03072
17	0.77491	0.05912	-1.46977	0.05424
18	0.71649	0.04024	-0.32155	0.03062

Shortcomings and Extensions.

There are some specifications made in the program that cannot be altered by a user. The priors on the parameters are fixed, that is, vague proper priors are used and they cannot be altered Fox (2002). The relations within the structural model are assumed linear with normal distributed noise. With a Metropolis-Hastings (MH) algorithm parameters of a model with non-linear relations and/or non-gaussian noise could be estimated.
The program does not do much about checking the input. There is a way to check the input with callback functions but this has not been implemented completely. So, the program will crash when illegal options are specified. It will shut down S-Plus with a *Fatal Error.* There is no escape possible, not saved results or script files will be gone. Then, it is also possible that the .Random.seed object within the work directory has been corrupted. It should be removed by giving the command rm(.Random.seed). Tip, before starting to compute be aware of the fact that S-Plus may shut down.
The scale of the latent independent variables are fixed to a standard normal scale and this cannot be changed. It should be possible in the future to identify a measurement model in different ways.
Callback functions should make it possible to fill in the necessary fields instead of all fields. This should be fixed quite fast. The output is structured but the headings switches sometimes, and tabs are not always on the right places, this really irritating work to correct. But sometimes,… There are other features, aspects to be mentioned later on.

Remove/De-Install

The software is removed by deleting the files S.dll and basicmlirt_160702 and typing in the command window of S-Plus;
rm(readdatalev12)
rm(dataindi)
rm(mlirtcallback2)
rm(mlirt2)
rm(resultsmlirt2)
rm(mlirtdialog2)
rm(printF)
rm(formatF)

rm(HPD).

References

References regarding the multilevel IRT software:

Fox, J.-P. (2001). Multilevel IRT: *A Bayesian perspective on estimating parameters and testing statistical hypotheses.* Unpublished doctoral dissertation, Twente University, Enschede, Netherlands.

Fox, J.-P. (2002). Multilevel IRT with dichotomous and polytomous items. Submitted for publication.

Fox, J.-P. (2003). Stochastic EM for Estimating the Parameters of Multilevel IRT Model. *British Journal of Mathematical and Statistical Psychology, 56,* 65-81.

Fox, J.-P., & Glas, C.A.W. (2001). Bayesian estimation of a multilevel IRT model using Gibbs sampling. *Psychometrika, 66*,269-286.

Fox, J.-P., & C.A.W. Glas. (2002). Modelling Measurement Error in Structural Multilevel Models. In G.A. Marcoulides and I. Moustaki (Eds.), *Latent Variable and Latent Structure Models* (pp. 245-269). London: Lawrence Erlbaum Associates, Publishers.

Fox, J.-P., & Glas, C.A.W. (2003). Bayesian Modeling of Measurement Error in Predictor Variables using Item Response Theory. *Psychometrika*, to be published.

Appendix

The object mlirtout is a list with the following elements;
mlirtout[[1]] : items scores of the latent (dependent) variable(s) at Level 1.
mlirtout[[2]] : sampled augmented data at the last iteration.
mlirtout[[3]] : input values of the discrimination parameter(s), corresponding to the irt model(s) at Level 1.
mlirtout[[4]] : input values of the difficulty parameter(s), corresponding to the irt model(s) at Level 1.
mlirtout[[5]] : input/starting values of the latent variables at Level 1.
mlirtout[[6]] : specifications of the polytomous and/or dichotomous data.
mlirtout[[7]] : items scores of the latent variable(s) at Level 2.
mlirtout[[8]] : sampled augmented data at the last iteration.
mlirtout[[9]] : input values of the discrimination parameter(s), corresponding to the irt model(s) at Level 2.
mlirtout[[10]] : input values of the difficulty parameter(s), corresponding to the irt model(s) at Level 2.
mlirtout[[11]] : input/starting values of the latent variables at Level 2.
mlirtout[[12]] : specifications of the polytomous and/or dichotomous data.

mlirtout[[13]] : specifies the number of latent variables at each level.
mlirtout[[14]] : specifies the number items corresponding to the different IRT models.
mlirtout[[15]] : optional observed dependent variable.
mlirtout[[16]] : observed explanatory variables with random coefficents at Level 1.
mlirtout[[17]] : observed explanatory variables with fixed coefficents at Level 1.
mlirtout[[18]] : observed explanatory variables at Level 2.
mlirtout[[19]] : starting value Level 1 variance.
mlirtout[[20]] : starting value Level 2 (co)variance matrix.
mlirtout[[21]] : starting values random coefficents Level 1.
mlirtout[[22]] : starting values fixed coefficients Level 1.
mlirtout[[23]] : starting values fixed coefficients Level 2.
mlirtout[[24]] : vector with the number of observations per group.
mlirtout[[25]] : specifying the fixed and random regression coefficients.
mlirtout[[26]] : relating Level 2 variables to random coefficients.
mlirtout[[27]] : number of Level 1 units.
mlirtout[[28]] : number of Level 2 units.
mlirtout[[29]] : random number.
mlirtout[[30]] : number of iterations.
mlirtout[[31]] : Sampled values of the fixed coefficients (Level 2). To obtain them in a specified order, fixedcoeffients = matrix(mlirtout[[31]],ncol=number of fixed coefficients,nrow=mlirtout[[30]]) This way, first the intercepts of all random coefficients are given. Subsequently, the coefficients from the observed explanatory variables and then the coefficients from the latent explanatory variables are given. First the coefficients relating to the random intercept (relating to observed and latent variables), then the next random coefficient according to equation one. This order is also present in the report file where the parameter estimates are given.
mlirtout[[32]] : sampled values of the Level 1 variance.
mlirtout[[33]] : sampled values of the fixed coefficients at Level 1. Lev1fixedcoeffients = matrix(mlirtout[[33]],ncol=number of fixed coefficients at Level 1,nrow=mlirtout[[30]]). Again, the first column denotes a fixed intercept or the coefficient of the first entered observed covariate at Level 1 with a fixed slope. After the coefficients of the observed covariates follow the coefficients of the latent variables with a fixed slope in the same order as they were filled in. This order equals the order in which the coefficients are printed in the report file.
mlirtout[[34]] : a matrix with dimension (8 + number of latent variables) columns and mlirtout[[30]] rows. Column one gives the number of trials if negative discrimination parameters were sampled corresponding to IRT model(s) at Level 1 at each iteration. If the value 20 occurs, the negative values has been replaced with one, in case of polytomous data 100 attempts are allowed before setting the discrimition values at one. Column two corresponds in the same way to IRT model(s) at Level 2. Column 3 captures information concerning the log-likelihood of the structural multilevel model after the burnin period. Column six contains sampled values of the inverse of the variance concerning the first random coefficient, usually the random intercept. Column 7 contains sampled values of the first random coefficient, usually the intercept, regarding the first group. Column 8 contains sampled values of the standard deviation of the disturbance at Level 1. For the first latent variable, column 9 contains the following information: the

first element presents the pseudo bayes factor. The second element, the number of times predictive data are sampled for posterior predictive checking and the fourth element contains the number of positive counts corresponding to the Chi-Square Discrepancy. The other values stored in this column concerns the computation of the other p-value. In case of a second latent variable, column 10 presents comparable output only based on a different indictor variable.

mlirtout[[35]] : sampled values of the random coefficients. They are stored in the following way. The command, randomcoeff <- matrix(mlirtout[[35]],ncol= #random coefficients,nrow=mlirtout[[30]]) will place all values of the first random coefficient, the random intercept, in the first column, the next random coefficient in the second column etc.

mlirtout[[36]]] : sampled values of the covariance matrix at Level 2, sampledT <- matrix(mlirtout[[36]],ncol=#elements in the covariance matrix,nrow=mlirtout[[30]]). Then, each row represent a draw, the first elements correspond to the first row of the covariance matrix in the same order, the second set of elements corresponds to the second row of the covariance matrix, etc. Say, T of dimension Q,

Iteration number	Covariance Matrix
1	$\sigma_{11}^2, \sigma_{12}^2, \ldots, \sigma_{1Q}^2, \sigma_{21}^2, \sigma_{22}^2, \ldots, \sigma_{2Q}^2, \ldots, \sigma_{QQ}^2$
2	$\vdots$
$\vdots$	$\vdots$

mlirtout[[37]] : EAP estimates of the latent variables at *Level 1*. In the same order as they were defined. EAP_LatentVariables_level1 <- matrix(mlirtout[[37]],ncol=#number of latent variables, nrow=mlirtout[[27]]). If the dependent variable is defined as latent the first column contains its expected a posteriori estimate.

mlirtout[[38]] : EAP estimates of the latent variables at *Level 2*. In the same order as they were defined. EAP_LatentVariables_level2 <- matrix(mlirtout[[38]],ncol=#number of latent variables, nrow=mlirtout[[28]]).

mlirtout[[39]] : Sampled values of the discrimination parameters. Again they are stored in the same order as defined. To obtain all sampled values, matrix(mlirtout[[39]], nrow=mlirtout[[30]], ncol=total number of items regarding all IRT models). This matrix has the following structure, (in case of two IRT models):

Iteration number	First IRT modelel	Second IRT model
1	$a_{11}, a_{12}, \ldots, a_{1K_1}$	$a_{21}, a_{22}, \ldots, a_{2K_2}$
2	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$

where a_{ik} is a discrimination parameter corresponding to item k and IRT model i.

mlirtout[[40]] : Sampled values of the difficulty parameters. These are stored the same way, that is, the command matrix(mlirtout[[40]], nrow=mlirtout[[30]], ncol=total number of items regarding all IRT models) results in the matrix

Iteration number	First IRT modelel	Second IRT model

1	$b_{11}, b_{12}, \ldots, b_{1K_1}$	$b_{21}, b_{22}, \ldots, b_{2K_2}$
2	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$

With the apply function parameter estimates can be obtained. S-plus fills a matrix by columns as default, obviously, filling a matrix by rows will result in different outcomes. All input variables are stored as above block structure.