STUIRT

A Computer Program for **S**cale **T**ransformation under
Unidimensional **I**tem **R**esponse **T**heory Models
Version 1.0
September 2004

Seonghoon Kim and **Michael J. Kolen**
Iowa Testing Programs
The University of Iowa

Under item response theory (IRT), scale transformation methods are statistical tools necessary to develop a common scale by finding a linear transformation between two scales that have been separately determined with different groups of examinees. Although several alternative IRT scale transformation methods have been developed, the following four have been most often described in the literature and used in practical testing programs: the mean/sigma (Marco, 1977), mean/mean (Loyd & Hoover, 1980), Haebara (Haebara, 1980) and Stocking-Lord (Stocking & Lord, 1983) methods. These methods were originally developed for dichotomous IRT models and have been extended to polytomous IRT models. Kim and Lee (2004) further extended the four methods to mixed-format tests. For the extension, as suitable *unidimensional* IRT models for each item format of a mixed-format test, the following five models were considered: the three-parameter logistic (3PL) model (Birnbaum, 1968), the graded response (GR) model (Samejima, 1969, 1972), the nominal response (NR) model (Bock, 1972), the generalized partial credit (GPC) model (Muraki, 1992), and the multiple-choice (MC) model (Thissen & Steinberg, 1984). Each of the four extended scale transformation methods is intended to handle mixed-format tests using any mixture of the five IRT models.

STUIRT is a computer program for implementing the four IRT scale transformation methods extended by Kim and Lee (2004) for use with mixed-format tests as well as single-format tests. STUIRT can be viewed as an extended version of the computer program POLYST (Kim & Kolen, 2003), which can handle only single-format test forms. As assumed in Kim and Lee (2004), STUIRT should be used when the constructs measured by

different formats of items are similar enough that some or all of the five unidimensional IRT models can be used together to analyze response data from mixed-format tests. The four scale transformation methods assume that a set of common items provides the link between two samples of data that are obtained from separate administrations of two forms (old and new) of a mixed-format test. The main task of each scale transformation method is to find the slope (*S*) and intercept (*I*) of a linear transformation relating the two scales that are separately calibrated in each sample.

Technically, STUIRT uses multivariate search techniques described in Dennis and Schnabel (1996) to find the transformation constants (i.e., *S* and *I*) that minimize the criterion functions for the Haebara and Stocking-Lord methods. The multivariate search techniques require partial derivatives of the criterion functions with respect to *S* and *I* to be evaluated. Kim and Kolen (in press) provide numerical details of the partial derivatives under each of the five IRT models. There is no limitation in the number of common items and the number of response categories for an item.

For users' information, advantages of STUIRT can be fully taken when it is used together with the computer program POLYEQUATE (Kolen, 2004), which is geared to test equating using a mixture of unidimensional IRT models. POLYEQUATE assumes that item and proficiency parameter estimates from old and new forms of a test are placed on a common scale. When the test data analyzed are obtained in the context of the common-item nonequivalent groups equating design (Kolen & Brennan, 2004), assuming the old form scale is a reference scale, item and proficiency parameter estimates on the new form should be placed onto the old form scale. STUIRT is typically used for this kind of scale transformation, so that a common IRT scale is developed. To run POLYEQUATE, several input files must be prepared in advance. By using the option keyword **`KO`**, which is described in detail later, users can create some of the input files easily.

Program Operation

STUIRT is a command-line program (i.e., console version) and thus it must be run using a command prompt window. A file named `stuirt.exe` is the executable file. The command line used to run the STUIRT program contains up to two arguments:

```
STUIRT input-file [output-file]
```

The first and second arguments are for the names of input and output files, respectively.

The second argument is optional (indicated by brackets) and must be distinct from the first argument; if the second argument is not specified, the resulting output file name is made of concatenating the input file name and a chosen extension `out`. Users can use up to 500 characters to name input and output files. Input files are often referred to as command files. The input file does not need to be in the same folder in which the executable file exists. How to create an input file is explained in the next section. If users run the program without arguments, a command prompt will ask them to enter two file names for input and output. Users should enter proper input- and output-file names, as the command prompt asks to do so. When the program is successfully executed, a message will show up on the screen to indicate where outputs are saved.

Creating Input Files

STUIRT has syntax rules for an input file. Detailed explanations for the rules are presented below. STUIRT expects the input file to be a text only file. For the input file it is strongly suggested that white space between keywords or input parameters should be generated typing one or more spaces rather than using tabs.

STUIRT Syntax Rules for Input Files

STUIRT has four main keywords and nine option keywords. The four main keywords are **`NE`**, **`OL`**, **`CI`**, and **`BY`**. The nine option keywords are **`ST`**, **`IT`**, **`ND`**, **`OD`**, **`FS`**, **`SY`**, **`LM`**, **`OP`**, and **`KO`**. Subkeywords follow some of the keywords.

There is one important principle applied in preparing input files. Every piece of the input information such as keywords, data, and delimiters must be separated by space(s) or new line escape character(s). *Never use commas to separate between pieces of the input information!*

An input file must begin with the keyword **`NE`** and end with the keyword **`BY`**. However, users can give comments by using a C-like comment style before the keyword **`NE`**. That is, the usage of */* preceding the body of comments followed by */* permits users to comment their input files. Data after the keyword **`BY`** are also ignored by STUIRT.

Below is a syntax diagram for the input file. Every keyword consists of two capital letters. Enter keywords as shown, although users can use lowercase or more than two letters. The syntax for each option keyword is provided separately in the last part of this section.

After this, specific and detailed explanation for each keyword will be given. The following conventions are applied in the syntax diagram:

Bold-faced Capital letters
indicate the exact same spelling and form as shown should be used.

Italic Small letters
indicate that specific information should be supplied by users.

Vertical bar (|) separating subkeywords and options
indicates only one of the terms separated by it should be chosen.

Bracketed information ([ ])
is optional.

Braces ({ })
indicate that the inside statements should be dealt with as a group.

Three periods (...)
indicate that the same kind of data are entered.

Syntax Diagram for Input Files

```
/*
 COMMENTS
*/
```

NE *number-of-items*
item-information-and-parameters-list
...
item-information-and-parameters-list

OL *number-of-items*
item-information-and-parameters-list
...
item-information-and-parameters-list

CI *number-of-items* { **AO** } |
{ **MA** *pair-of-new-and-old-items-list* }

[OPTION]
...
[OPTION]

BY

Syntax Diagram for Options

ST *slope intercept*

IT *number-of-iterations*

OD *n* { **GH** } |

{ **PN** *mean std multiple-of-std* } |

{ **PB** *alpha beta lower-limit upper-limit* } |

{ **RN** *mean std* } |

{ **RU** *lower-limit upper-limit* } |

{ **ED** *starting ending* **/ EQ |** { **WL** *theta-weight-list* } } |

{ **SE** {**DI** *pair-of-value-and-weight-list* } |

{**FI** *filename* } }

ND *the same as* **OD**

FS { **DO** | **NO** } { **DO** | **NO** }

SY { **BI** | **NO** | **ON** } { **BI** | **NO** | **ON** }

LM *slope intercept*

/ *number-of-searches radius tolerance* **NO** | **IN** | { **FI** *filename* }

OP

KO **MM** | **MS** | **HA** | **SL**

Specific explanations for keywords are presented below.

NE *number-of-items*
item-information-and-parameters-list
. . .
item-information-and-parameters-list

The keyword **NE** stands for a **new** form, whose item parameter estimates should be placed on the scale determined by the old form. The keyword **NE** is followed by a number of items on the new form. Some of the items serve as common items, through which the new-to-old scale transformation is conducted. A record for each item is typed differently according to the IRT model considered as follows [the notation for parameter estimates of item *j* having K_j categories is borrowed from Kim and Kolen (in press)]:

• The 3PL Model

ID_j **L3** *2* **DW** | { **SW** U_{j1} U_{j2} } D $\hat{a}_j$ $\hat{b}_j$ $\hat{c}_j$

• The GR Model

ID_j **GR** K_j **DW** | { **SW** $U_{j1} \cdots U_{jK_j}$ } D $\hat{a}_j$ $b_{j2} \cdots b_{jK_j}$

• The GPC Model

ID_j **PC** K_j **DW** | { **SW** $U_{j1} \cdots U_{jK_j}$ } D $\hat{a}_j$ { **IS** $b_{j1} \cdots b_{jK_j}$ } |
{ **LC** $\hat{b}_j$ $d_{j1} \cdots d_{jK_j}$ }

• The NR Model

ID_j **NR** K_j **DW** | { **SW** $U_{j1} \cdots U_{jK_j}$ } $\hat{a}_{j1} \cdots \hat{a}_{jK_j}$ $b_{j1} \cdots b_{jK_j}$

• The MC Model

ID_j **MC** K_j **DW** | { **SW** $U_{j0} \cdots U_{jK_j}$ } $\hat{a}_{j0} \cdots \hat{a}_{jK_j}$ $b_{j0} \cdots b_{jK_j}$ $d_{j1} \cdots d_{jK_j}$

The first input component ID_j indicates an **identification** number of item j. The identification number is used when matching common items on the new and old forms. The second keywords **L3**, **GR**, **PC**, **NR**, and **MC** stand for the 3PL, GR, GPC, NR, and MC

models, respectively. The next integer K_j indicates the number of response categories of item j. The two subkeywords, **DW** and **SW**, are used 1) to assign scoring weights (which are program- or user-supplied) to response categories and/or 2) to select response categories that are used for estimating the slope and intercept of the new-to-old linear transformation. The subkeyword **DW** stands for the **default** scoring **weights** and stands alone without being followed by a list of scoring weights. Scoring weights are necessary to define a true score of an examinee at a given value of proficiency. For the 3PL, GR, and GPC models, the default scoring function is $k-1$, where k is the category index; specifically, for example, in the case of a 4-category item, a list of 0 1 2 3 is used to provide response categories with scoring weights. In contrast, under the NR and MC models, the true score typically is not defined; thus, the default scoring function for the NR model consists of K_j 0's and the MC model uses K_j plus one 0's to define the default scoring function.

On the other hand, **selection** of response categories and/or assignment of scoring **weights** are made with the subkeyword **SW**. A list of scoring weights must follow **SW**; as suggested by the default scoring functions, K_j weights are needed for items under the 3PL, GR, GPC, and NR models and K_j plus one weights under the MC model. The default setting used by the program states that all respone categories contribute to the estimation of the slope and intercept of the new-to-old transformation. The "0" or "Don't Know" latent category in the MC model, however, is involved only when estimating transformation constants from the mean/mean and mean/sigma methods. This default setting can be overriden through the use of the special subkeyword **NA** (not applicable). Typing **NA**'s instead of numerical values of scoring weights for any response categories of item j prevents those response categories from being involved in the estimation of transformation constants. To deselect the first and fourth response categories of a 4-category item, for example, the following list of scoring weights may be used: **NA** 1 2 **NA**. In the case of common items, the deselction affects both new and old forms. Because specifications of the common items are determined by the new form, if response categories on the new form are deselected, the counterparts on the old form are deselected. Note that when deselection occurs, the deselected categories keep the values set by the default scoring function, $k-1$, as their scoring weights.

After assigning scoring weights and/or selecting response categories desired, a scaling constant (denoted as *D*, which is typically either 1.7 or 1.0) and/or item parameter estimates must be provided properly by model. Users should make sure two points for this provision.

First, item j under the GR model has $K_j - 1$ difficulty (threshold) estimates, ranging from $\hat{b}_{j2}$ to $\hat{b}_{jK_j}$. Second, the two subkeywords, **IS** and **LC**, are used to designate the type of item-category parameters under the GPC model. The subkeyword **IS** stands for "item-step" parameters and the subkeyword **LC** "location and category" parameters. Under the GPC model, an item-step parameter b_{jk} for category *k* of item *j* is often resolved into two parameters b_j (location) and d_{jk} (category) as follows: $b_{jk} = b_j - d_{jk}$. Note that the first item-step parameter estimate $\hat{b}_{j1}$ (typically 0) and the first category parameter estimate $\hat{d}_{j1}$ (typically 0) must be supplied by users.

OL *number-of-items*
item-information-and-parameters-list
...
item-information-and-parameters-list

The keyword **OL** stands for an **old** form, which is assumed to be the reference form. Pieces of information following the keyword **OL** are typed in the same manner as in the keyword **NE**.

CI *number-of-items* { **AO** } |
{ **MA** *pair-of-new-and-old-items-list* }

The keyword **CI** is used to designate **items** that are **common** to both new and old forms. To specify the number of common items, an integer must follow **CI**. The two subkeywords **AO** (already ordered) and **MA** (match) are used to designate the common items that serve as the link between the new and old forms. If the first *number-of-items* items on the new form correspond, in order and as common items, to the first *number-of-items* items on the old form, the subkeyword **AO** can be used. In this case, make sure that pieces of information for a common item should be entered at the same position in both **NE** and **OL** inputs. Otherwise, the subkeyword **MA** can be used to designate common items on the new form and the corresponding items on the old form. Suppose that there are two common items and the first and second items on the new form match with the third and fourth items on the old form. Pairs of items, then, must be designated as follows:

```
CI 2 MA
1 3
```

2 4
Make sure that, in a pair of two numbers, the first one is for the new form and the second one corresponds to the old form.

As described earlier, specifications of the common items are determined by the new form. The specifications of concern include (1) category selection scheme, (2) scoring function, and (3) scaling constant. (If the new and old forms have a different model specification and/or a different number of response categories for a common item, STUIRT will give a relevant error message and stop.) Thus, for example, if response categories on the new form are deselected, the counterparts on the old form are deselected. Because the new and old forms are expected to have the same specifications for the common items, this issue might not be critical. However, the same specfications for the common items must be used to avoid wrong solutions for the slope and intercept of the new-to-old transformation.

[OPTION]
...
[OPTION]

As described above, there are nine option keywords to accommodate users' needs. Most of the option keywords are used to control the solutions of the Haebara and Stocking-Lord methods. The order of the option keywords does *not* matter. The detailed explanations for usage are given below.

BY

The keyword **BY** (bye) stands alone. An input file must end with **BY**. STUIRT ignores all input data after the keyword **BY**. Therefore, users can write their additional comments after this keyword.

[OPTION KEYWORDS]

ST *slope intercept*

The keyword **ST** is used to provide **starting** values for the slope and intercept of the linear transformation in the Haebara and Stocking-Lord methods. Users should supply two real

numbers for the slop and intercept in order. The default values of the slope and intercept are 1.0 and 0.0, respectively, unless this option keyword is used.

IT *number-of-iterations*

The keyword **IT** is for the maximum number of **iterations** to obtain transformation constants that minimize criterion functions for the Haebara and Stocking-Lord methods. Users should supply an integer they want for the iteration number. The default maximum number of iterations is 20.

OD *n* { **GH** } |
{ **PN** *mean std multiple-of-std* } |
{ **PB** *alpha beta lower-limit upper-limit* } |
{ **RN** *mean std* } |
{ **RU** *lower-limit upper-limit* } |
{ **ED** *starting ending* **/ EQ** **|** { **WL** *theta-weight-list* } } |
{ **SE** {**DI** *pair-of-value-and-weight-list* } |
{**FI** *filename* } }

ND *the same as* **OD**

The keyword **OD** (old group's distribution) is used to specify the proficiency distribution (characterized by proficiency points and weights) of the old group of examinees, so that criterion functions for the Haebara and Stocking-Lord methods can be defined. Likewise, the keyword **ND** (new group's distribution) is used to provide STUIRT with the proficiency distribution of the new group of examinees. The integer *n* after **OD** or **ND** is the number of proficiency points and should be a number between 1 and 1000, inclusive. When the option keyword **SY** is used with its subkeyword **BI**, the proficiency points and weights provided by the both **OD** and **ND** keywords are used. When the option keyword **SY** is used with its subkeyword **NO**, only the proficiency points and weights provided by the keyword **OD** are used; with the subkeyword **ON**, only the **ND** inputs are used. For more information about the criterion functions defined on each of new and old scales, refer to Kim and Lee (2004). The usage of **ND** and that of **OD** are the same except that proficiency points and weights are assigned separately on each of new and old scales.

Specifically, the criterion functions for the Haebara and Stocking-Lord methods are

subject to the summation (or integration) scheme over examinees to incorporate the proficiency distributions on the old and new scales. Kolen and Brennan (2004) present several ways to specify the proficiency points and their weights to be incorporated into the criterion function in question. The **OD** and **ND** support all the ways described in Kolen and Brennan (2004). Each of **OD** and **ND** has seven subkeywords: **GH**, **PN**, **PB**, **RN**, **RU**, **ED**, and **SE**.

First, **GH** stands for the Gauss-Hermite quadrature points and weights. This subkeyword can be used properly, if a continuous distribution of proficiency is known or estimated and the summation of the criterion function could be replaced by integration over the proficiency continuum. In the program, the proficiency distribution is assumed a standard normal one. The possible maximum number of quadrature points is 180. Although more than 180 quadrature points are theoretically possible, the authors' experiences suggest that quadrature weights tend to be unstable when trying to obtain more than about 200 quadrature points. According to Zeng and Kolen (1994), even 80 quadrature points seem to be enough to estimate the slope and intercept of a linear transformation to a satisfactory degree.

Second, **PN** stands for a **polygonal** approximation to a **normal** distribution. A polygonal approximation is often encountered in finding areas and evaluating integrals. **PN** can be used to evaluate *n* proficiency points and their weights to approximate a normal distribution having a value of *mean* and a value of *std*, with a left end point and a right end point being $mean - multiple\text{-}of\text{-}std \times std$ and $mean + multiple\text{-}of\text{-}std \times std$, respectively. More specifically, *n* proficiency points are equally spaced over the range of a left end point to a right end point. At each proficiency point, the density from *N(mean, std)* is computed. All the densities are summed and then each density is divided by the sum so that the densities are standardized. The resulting values of densities are used as the weights. These steps are similar to those used in PC-BILOG (Mislevy & Bock, 1986).

Third, **PB** stands for a **polygonal** approximation to a four-parameter **beta** distribution $Beta(\alpha, \beta, l, u)$, where α and β are two scale parameters and l and u are lower and upper limits. To evaluate *n* proficiency points and their weights, the same logic used in **PN** is applied except that the interval [*l, u*] is divided into *n* subintervals and then the midpoint of each subinterval is used for a proficiency point.

Fourth, **RN** stands for **random** numbers from a **normal** distribution. With two real numbers for a mean and a standard deviation, **RN** generates *n* pseudo-random proficiency

values sampled from a normal distribution, *N(mean, std)*.

Fifth, **RU** stands for **random** numbers from a **uniform** distribution. With two real numbers for a lower limit and an upper limit, **RU** generates *n* pseudo-random proficiency values ranging from *lower-limit* to *upper-limit*.

Sixth, **ED** stands for **equal distance** in the intervals between two theta points on the proficiency scale. Users should supply two real numbers for a starting point and an ending point. The theta continuum ranging from the starting point to the ending point is divided into $n-1$ intervals with an equal length. Two subkeywords, **EQ** and **WL**, preceded by a slash are prepared to provide weights associated with their theta values. **EQ** stands alone and is used to give the same constant weight 1.0. **WL** is used to list weights associated with their theta values, and the weights should be supplied with values users want.

Seventh, **SE** is used to specify *n* **selected** proficiency values and their corresponding weights in pairs. The two subkeywords, **DI** and **FI**, must follow **SE**. When using **DI**, selected proficiency values and their corresponding weights are input as follows.

```
-2.0 0.001
-1.5 0.015
 ...
 1.5 0.015
 2.0 0.001
```

When selected proficiency values and their weights are read in from an outside file, **FI** is used and the input format must be the same as that used in **DI**. The output file does not need to be located in a folder in which the executable file exists.

The default setting for proficiency values and their weights is that 25 proficiency values, which are equally spaced between -3.0 and 3.0, are used with the same weight 1.0 for all proficiency values. The default setting can be expressed as 25 **ED** -3.0 3.0 / **EQ**.

FS { **DO** | **NO** } { **DO** | **NO** }

The option keyword **FS** is used for **standardizing** the criterion **functions** for the Haebara and Stocking-Lord methods. The two subkeywords, **DO** and **NO**, are used to specify options. The first **DO** or **NO** is for the Haebara method and the second **DO** or **NO** is for the Stocking-Lord method. **DO** means that standardization is done and **NO** means that no standardization is done. What is meant by standardization of the criterion function is that one divides a sum of squared differences between characteristic curves in the criterion function by the number

of the squared differences or the sum of weights assigned to the differences. For example, usually, to standardize the criterion function for the Stocking-Lord method, one divides the sum of squared differences between test characteristic curves by the number of proficiency values (or examinees). For more detailed information, refer to Kim and Lee (2004).

Theoretically, the standardization of the criterion functions should not affect the solutions of the Haebara and Stocking-Lord methods. However, in practice, it could affect the solutions since the minimization algorithm used for nonlinear problems is affected by the magnitude of the criterion function due to its stopping rules. By default, standardization is conducted for both the Haebara and Stocking-Lord criterion functions.

SY { **BI** | **NO** | **ON** } { **BI** | **NO** | **ON** }

The option keyword **SY** is used to define criterion functions as non-symmetric or symmetric. Three subkeywords, **BI**, **NO**, and **ON**, are used to specify options. The first **BI**, **NO**, or **ON** is for the Haebara method and the second **BI**, **NO**, or **ON** is for the Stocking-Lord method. Theoretically, the criterion function for either of the Haebara and Stocking-Lord methods could be defined in three symmetry-related ways. The first is one in which the criterion function is defined only on the old scale as in the typical use of the Stocking-Lord method (new-to-old direction: **NO**). The second is one in which the criterion function is defined only on the new scale (old-to-new direction: **ON**). The third is one in which the criterion function is defined on the both old and new scales as in the use of the Haebara method (new-to-old and old-to-new, i.e., bi-directional: **BI**)

Theoretically, the three ways, **BI**, **NO**, and **ON**, to define the criterion function in question should give the same solutions as far as sampling error and model misfit do not happen. However, with sample data, the three ways will give different solutions for scale transformation. The default setting is in such that **SY BI BI**.

LM *slope intercept*
/ *number-of-searches radius tolerance* **NO** | **IN** | { **FI** *filename* }

The option keyword **LM** is used to search for possible **local minimum** solutions for the scale transformation constants after the first solutions for the Haebara and Stocking-Lord methods are obtained. Three subkeywords, **NO**, **IN**, and **FI** are prepared to instruct the program how and where to show the resulting history of local minimum search. If **NO** is

used, no history is shown in an output file. If **IN** is used, the resulting history is shown within the main output file specified by users or opened by the program. If **FI** followed by a file name is used, the resulting history is saved separately in the file, which does not need to be located in the folder having the executable file of STUIRT.

The criterion function defined by either of the Haebara and Stocking-Lord methods is nonlinear with respect to the scale transformation constants, S and I. Nonlinear systems could yield a local minimum but not a global minimum. To make sure that the obtained solutions (slope and intercept) are globally minimal, it is suggested that the option keyword **LM** be used. The first two real numbers, *slope* and *intercept*, are the values of the slope and intercept used to designate a center point of search. The center point is fixed and used for all iterations of local minimum search. An integer value, *number-of-searches*, after a slash is the number of iterations for local minimum search. The number of iterations should range from 0 to 1000. A real number, *radius*, is a radius for local minimum search. With the center point being an origin, *number-of-searches* pseudo-random points within the radius are selected. If a selected point (slope, intercept) is within the radius but the value of slope is negative, the *absolute* value of slope is used. Then, the resulting point is used as the starting point for each of iterations of local minimum search. The last real number, *tolerance*, represents a tolerance limit for each of the slope and intercept, within which all solutions for the slope and intercept are regarded as the same. For example, suppose that the tolerance limit is given as 0.01. If the solution of the slope at the first search is 1.21 and the slope at the second search is 1.23, two solutions are regarded as different from each other and, in turn, the local minimum points from the two searches are regarded as different from each other. In this case, a * will appear in a table that presents a history of the local minimum search.

OP

The option keyword **OP** is used when users want to see in detail how options were given to the program. **OP** stands alone. Without **OP**, the section [OPTIONS AND DEFAULTS] will not show up in a main output file.

KO { **MM** | **MS** | **HA** | **SL** }

The option keyword **KO** is used to create input files for the program POLYEQUATE (Kolen,

2004), which conducts IRT true and observed score equating using dichotomous and polytomous IRT models. As described earlier, POLYEQUATE assumes that item and proficiency parameter estimates from old and new forms of a test are placed on a common scale. To place item parameter estimates from the new form onto the old scale, one of the four scale transformation methods must be specified. The four subkeywords, **MM**, **MS**, **HA**, and **SL** stand for the mean/mean, mean/sigma, Haebara, and Stocking-Lord methods, respectively. By default, the Haebara method is used. POLYEQUATE runs based on five input files. With this option keyword, some of the input files can be prepared by copying data provided at the end of the main output file from STUIRT. Under the output title "===[Inputs for Kolen's POLYEQUATE]===", input data for the new form, for the old form, and for the transformed proficiency points and weights for the new form are provided.

Errors in Running STUIRT

STUIRT gives a specific error message when it does not understand the syntax in the input file. The error message will show up on the screen and tell possible sources of the error. Based on the message shown on the screen, users could find out any syntax errors in their input files of STUIRT. Users could also get a hint by opening the output file, which is not successfully closed. In most cases, the last part of the output file will give a hint enough to guess where a syntax error happened.

Reading Output Files

Reading an output file is straightforward. The output file first replicates the contents of the input file, and so users can verify if the data were read properly. If the option keyword **OP** were used, the options given to the program would show up. Then, means and standard deviations of the common item parameter estimates are presented, along with the solutions for the moment methods (i.e., the mean/mean and mean/sigma methods). Next, the first solutions for the characteristic curve methods (i.e., the Haebara and Stocking-Lord methods) are presented. If the local minimum search were tried by the option keyword **LM**, a history of the search would show up. Then, all solutions by method are summarized again and, subsequently, how to read termination codes ranging from 1 to 5, which are passed from the minimization algorithm used in the program, is provided to give users information

about the characteristics of the solutions of the Haebara and Stocking-Lord methods. If the program generates any termination code other than 1, it is strongly suggested that users try again with different starting values for the solutions or with the **LM** option. Finally, if the option keyword **KO** is used, input data for the computer program POLYEQUATE are provided.

Notification of Authors

It is requested that users of STUIRT provide Seonghoon Kim or Michael J. Kolen with their email addresses by sending them a brief email message at

seonghoon-kim@uiowa.edu or michael-kolen@uiowa.edu.

This will permit contacting users about any errors, additions, or explanations relevant to the program.

Distribution

STUIRT may be distributed to others without obtaining the permission of the authors. However, it is requested that anyone who uses this program contact the authors in the manner indicated above.

Disclaimer

No warranties are made that STUIRT is free of error, that it is consistent with any particular standard, or that it will meet the requirements of any particular application. The authors disclaim any direct or consequential damages resulting from use of this program.

References

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F.M. Lord and M.R. Novick (Eds.), *Statistical theories of mental test scores* (pp. 397-472). Reading, MA: Addison-Wesley.

Bock, R.D. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika, 37*, 29-51.

Dennis, J.E., & Schnabel, R.B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*. Philadelphia: Society for Industrial and Applied Mathematics.

Haebara, T. (1980). Equating logistic proficiency scales by a weighted least squares method. *Japanese Psychological Research, 22*, 144-149.

Kim, S., & Kolen, M.J. (2003). *POLYST: A computer program for polytomous IRT scale transformation*, Iowa City, IA: Iowa Testing Programs, The University of Iowa. (Available from the web address: http://www.uiowa.edu/~casma)

Kim, S., & Kolen, M.J. (in press). *Methods for obtaining a common scale under unidimensional IRT models: A technical review and further extensions* (Occasional Paper). Iowa City, IA: Iowa Testing Programs, The University of Iowa.

Kim, S., & Lee, W. (2004). *IRT scale linking methods for mixed-format tests* (ACT Research Report 2004-5). Iowa City, IA: ACT, Inc.

Kolen, M.J. (2004). *POLYEQUATE: A computer program*. Iowa City, IA: The University of Iowa. (Available from the web address: http://www.uiowa.edu/~casma)

Kolen, M.J., & Brennan, R.L. (2004). *Test equating, scaling, and linking: Methods and practices* (2nd ed.). New York: Springer.

Loyd, B.H. & Hoover, H.D. (1980). Vertical equating using the Rasch model. *Journal of Educational Measurement, 17*, 179-193.

Marco, G.L. (1977). Item characteristic curve solution to three intractable testing problems. *Journal of Educational Measurement, 14*, 139-160.

Mislevy, R.J., & Bock, R.D. (1986). *PC-BILOG: Item analysis and test scoring with binary logistic models*. Mooresville, IN: Scientific Software Inc.

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement, 162*, 159-176.

Samejima, F. (1969). Estimation of a latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement, 17*.

Samejima, F. (1972). A general model for free response data. *Psychometrika Monograph Supplement, 18*.

Stocking, M., & Lord, F.M. (1983). Developing a common metric in item response theory. *Applied Psychological Measurement, 7*, 207-210.

Thissen, D., & Steinberg, L. (1984). A response model for multiple choice items. *Psychometrika, 49*, 501-519.

Zeng, L., & Kolen, M.J. (1994). *IRT scale transformations using numerical integration*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans.