

# **PIE**

## **A Computer Program for IRT Equating**

**Version 1.0**

**April 1995**

**Brad Hanson and Lingjia Zeng**

**American College Testing**

**P.O. Box 168, Iowa City, IA 52243**

This manual describes a program for equating tests using item response theory (IRT) methods. This program performs both IRT true score equating and IRT observed score equating (Lord, 1980). The basic IRT model used in the program is the three-parameter logistic model with  $D=1.7$  as described in Lord (1980).

### **Program Operation**

This manual describes the Macintosh version of PIE. This description of the program assumes familiarity with IRT item parameter scaling methodology and familiarity with basic Macintosh operations.

When the program begins only the menu bar is visible. To compute an equating select the "Compute Equating" item from the File menu. A standard file dialog is presented to select an input data file to read (the format of the input file is described below). After the input file has been selected a standard file dialog is presented for specifying an output file in which to write the equating results. The equating results are then computed and written to the output file. After the computations are complete another equating can be computed by selecting Compute Equating from the File menu, or the program can be terminated by selecting Quit from the File menu.

The output file produced by the program is a text file that can be read by any text editor or word processor. The output file contains the name of the input file from which the data were read, the IRT true score equating results, and IRT observed score equating results (observed score equating is only reported if a theta distribution is read). The sum of the  $c$  item parameters for the new form is reported (this determines the lowest new form score for which an equivalent old form score can be directly computed for true score equating). For each new form raw score four numbers are reported for the IRT true score equating – 1) the theta equivalent, 2) the true score equivalent on the old form, 3) the unrounded scale score equivalent, and 4) the rounded scale score equivalent. For each new form raw score three numbers are reported for the IRT observed score equating – 1) the raw score equivalent on the old form, 2) the unrounded scale score equivalent, and 3) the rounded scale score equivalent. In

addition, the fitted observed score distributions for the new and old form that are used to compute the observed score equating function are reported. The fitted observed score distributions are computed using the IRT item parameter estimates for the new and old forms and the theta distribution.

The procedures used to calculate the true and observed score equatings are described in Kolen and Brennan (1995). The routine RTSAFE from *Numerical Recipes in C* (Press, Teukolsky, Vetterling, & Flannery, 1992) was used in the calculation of the IRT true score equating function.

### **Format of Input Data File**

The input data file must contain item parameters for both the new and old forms. Optionally, the data file contains a theta distribution used to compute IRT observed score equating, and a raw to scale score transformation for the old form. If a theta distribution is not read, then IRT observed score equating is not computed. If no raw to scale score conversion for the old form is read then the old form raw to scale score transformation is assumed to be an identity function.

All item parameter estimates and the theta distribution read by the program must be on a common scale. The ST program (Hanson & Zeng, 1995) can be used to put two sets of item parameter estimates with common items on the same scale. An estimated theta distribution based on data used for the new form item parameter estimation must be transformed using the same scale transformation used on the item parameters for that form.

Each of the four possible pieces of information in the input data file (old and new item parameter estimates, theta distribution, and old form raw to scale score transformation) begins with a information line followed by the appropriate data. The information line consists of a dollar sign in the first column followed by one or more blanks or tabs and a string describing the information to follow.

An example used throughout this manual uses data from Chapter 6 of Kolen and Brennan (1995). Table 1 presents the contents of an input data file using data taken from Tables 6.8 and 6.10 of Kolen and Brennan (1995). The theta distribution used is for the group taking Form X.

The first eight lines contain a comment. The program will skip over these lines. The ninth line has a dollar sign as the first character. This indicates that this line is a information line preceding information to be read in by the program. The text "Item\_Parameters" after the dollar sign in the ninth line indicates that the new form item parameters will follow starting on the next line. Following the new form item parameters is a line beginning with a dollar sign which is followed by the text "Link\_Item\_Parameters". Following this line are the old form item parameters. Each line following the "Item\_Parameters" and "Link\_Item\_Parameters" information lines contains one string and three numbers: 1) an string identifying an item (maximum of eight characters); 2) the slope item parameter (a) corresponding to that item; 3) the threshold item parameter (b); and 4) the lower asymptote item parameter (c). The string and three numbers must be separated by one or more blanks or tabs.

**Table 1.**  
**Example Data File.**

This is an example input data file for the program PIE taken from Tables 6.8 and 6.10 "Test Equating: Methods and Practices" by Michael Kolen and Robert Brennan.

The new form (Form X) and old form (Form Y) item parameters are taken from Table 6.8. The theta distribution is taken from Table 6.10. It is the Form X theta distribution converted to the to the Form Y scale. An identify function is used for the old form (Form Y) raw to scale score conversion.

\$ Item\_Parameters

1	0.467344	-2.616539	0.175056
2	0.670924	-1.068342	0.116490
3	0.386976	-1.339358	0.208748
4	1.228024	0.064115	0.282599
5	0.828161	-0.701780	0.262510
6	0.496452	-1.511753	0.203834
7	0.731571	0.030429	0.322391
8	0.973113	-0.657195	0.220907
9	0.641447	-0.479277	0.159961
10	0.779740	0.688159	0.364807
11	0.815589	0.344648	0.239862
12	0.563967	-0.444704	0.123961
13	1.047915	-0.014127	0.253470
14	0.892128	0.422782	0.156932
15	0.908926	0.626040	0.298628
16	0.781655	0.213016	0.252099
17	0.759750	0.098919	0.227257
18	0.822383	-0.274926	0.053538
19	0.557961	-0.051057	0.120098
20	0.897580	0.610857	0.203621
21	0.295784	2.173474	0.148927
22	0.716916	0.742542	0.233231
23	0.947343	0.180908	0.064409
24	1.239634	0.700229	0.245270
25	0.436801	1.117559	0.142681
26	0.781728	0.763879	0.087920
27	1.599493	1.149537	0.199245
28	1.279241	1.270811	0.164220
29	0.821706	1.311960	0.143102
30	0.596854	2.130355	0.085290
31	1.075705	1.701967	0.244298
32	0.728471	1.511532	0.086474
33	1.197162	1.325326	0.078897
34	0.493857	3.580113	0.139884
35	0.787070	3.165387	0.108961
36	1.104752	2.034859	0.107530

\$ Link\_Item\_Parameters

1	0.870350	-1.450715	0.157647
2	0.462772	-0.406996	0.109378
3	0.441595	-1.334933	0.155883
4	0.544796	-0.901734	0.138071
5	0.619973	-1.486483	0.211368
6	0.572995	-1.321004	0.191298
7	1.175228	0.069050	0.294746
8	0.445023	0.232402	0.272323

**Table 1 (continued).  
Example Data File**

9	0.598719	-0.709831	0.117663
10	0.847924	-0.425342	0.144462
11	1.031996	-0.818383	0.093584
12	0.604125	-0.353942	0.081759
13	0.829722	-0.019137	0.128310
14	0.725171	-0.315511	0.085425
15	0.990164	0.531956	0.302443
16	0.774935	0.539442	0.217930
17	0.594230	0.898656	0.229885
18	0.808079	-0.115649	0.064791
19	0.964044	-0.194763	0.163258
20	0.783557	0.350592	0.129939
21	0.413973	2.553812	0.240967
22	0.761758	-0.158110	0.113708
23	1.195895	0.505649	0.239728
24	1.355437	0.581109	0.224322
25	1.186899	0.622889	0.257697
26	1.029556	0.389830	0.185611
27	1.041731	0.939158	0.165121
28	1.205473	1.135046	0.232287
29	0.969740	0.697642	0.107035
30	0.633562	1.896027	0.079396
31	1.082216	1.386423	0.185511
32	1.019458	0.919670	0.102719
33	1.134661	1.079013	0.063009
34	1.194845	1.841148	0.099913
35	1.196146	2.029683	0.083187
36	0.925521	2.133706	0.125873

\$ Theta\_Distribution

-5.2086	0.000101
-4.163	0.00276
-3.1175	0.03021
-2.072	0.142
-1.0269	0.3149
0.0184	0.3158
1.0635	0.1542
2.1090	0.03596
3.1546	0.003925
4.2001	0.000186

\$ Link\_Conversion

Form Y

37	0	36	1	1
0	0	0	0	
1	1	1	1	
2	2	2	2	
3	3	3	3	
4	4	4	4	
5	5	5	5	
6	6	6	6	
7	7	7	7	
8	8	8	8	
9	9	9	9	
10	10	10	10	
11	11	11	11	
12	12	12	12	

**Table 1 (continued).  
Example Data File**

13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36

The strings identifying the items are not used by the program but must still be present in the input data.

The optional information in the data file concerning the theta distributions and old form raw to scale score conversion is identified by the "Theta\_Distribution" and "Link\_Conversion" information lines. Each line under the Theta\_Distribution information line contains two numbers. These represent a set of quadrature points and weights for a theta distribution. The first number is the value for one quadrature point and the second number is the weight corresponding to that quadrature point. The quadrature points must be on the same scale as the item parameter estimates for the new and old forms.

The first line following the Link\_Conversion information line contains the form name of the form whose raw to scale score conversion follows (the form name is not used by the program). The second line contains five numbers describing the old form raw to scale score conversion: 1) the number of possible raw scores (number of items plus one), 2) the minimum scale score, 3) the maximum scale score, 4) the difference between consecutive scale scores (this must be constant for all adjacent scale scores), and 5) whether the raw to scale score conversion is linear (a one means the conversion is linear, a zero means it is not linear). The raw to scale score conversion begins on the third line. Each line beginning with the third line contains four numbers: 1) a raw score , 2) the unrounded scale score associated with the raw score, 3) a rounded and truncated version of the unrounded scale score, 4) the reported scale score (the reported scale score can differ from the rounded scale score if adjustments to the rounded raw to scale score conversion have been made). It is assumed that the raw scores for both

the old and new forms are number correct scores (integers from zero to the number of items on the test). The scale score properties for the new form are assumed to be the same as those read for the old form.

The four pieces of information in the input data file may appear in any order. For example, theta distribution could be before item parameters in the file. The input data file must at a minimum contain old and new form item parameters. If the input file does not contain a theta distribution then only the IRT true score equating function will be computed. If there is no old form raw to scale score conversion in the input file then the old form raw to scale score conversion is assumed to be the identity function.

## Scripting Support

Operation of PIE can be controlled via a scripting system such as AppleScript. This allows batch operation of the program. The two commands in the File menu (Computing Equating and Quit) have corresponding AppleScript commands. Documentation of the AppleScript commands is available by examining the scripting dictionary for PIE using the AppleScript Script Editor (the Script Editor is part of the standard AppleScript software). To view the dictionary for PIE select the Open Dictionary item from the file menu of the Script Editor. Select the PIE application icon in the standard file dialog that is displayed to open the PIE scripting dictionary. The entry from the dictionary concerning the **compute** command is given below.

**compute** file specification -- *File containing input data*  
          [**output file** file specification] -- *File output is written to*  
          [Result: string] -- *Text containing equating results*

The direct parameter of the **compute** command is a specification of the file containing the input data. There is an optional **output file** parameter that can contain a specification for an output file. If the **output file** parameter is not present the text containing the scaling results is returned by the compute command, otherwise an alias of the file containing the output text is returned. The following is an example of an AppleScript script for PIE using the **compute** command.

```
set infile to alias "Hard Disk:IRT equating:Kolen input"
set outfile to a reference to file "Hard Disk:IRT equating:Kolen out"
tell application "PIE"
    compute infile output file outfile
end tell
```

## Issues in Using the Program

PIE requires Macintosh system software version 7.0 or later. There are two versions of PIE - a version that will run on any Macintosh (a "fat" version) and a version for 680X0 Macintoshes with a math coprocessor (an FPU version) . The fat version runs on both Macintoshes with 680X0 processors

and Macintoshes with PowerPC processors (the program runs native on Power Macintoshes). The FPU version will only run on Macintoshes with 680X0 processors and a math coprocessor. The FPU version of the program will run considerably faster than the fat version on 680X0 Macintoshes that have a math coprocessor.

There are no limits on the number of items parameters or theta values read, other than the amount of memory allocated to the program. The memory partition might need to be increased for tests with large numbers of items. To change the memory partition select the PIE application in the Finder and choose "Get Info" from the File menu. Editing the text next to "Current size" will change the amount of memory the program uses.

The program will report an error if the input file is not in the correct format. Things to check for if the program reports an error in reading the input data are: 1) both new form and old form item parameters must be present, 2) the first character on an information line must be a dollar sign (there cannot be spaces before the dollar sign), 3) the text identifying the information to be read on an information line is case sensitive (it must match exactly), 4) the strings identifying the items can not be longer than 8 characters.

## References

Kolen, M. J., & Brennan, R. L. (1995). *Test Equating methods and practices*. New York: Springer-Verlag.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hanson, B. A. & Zeng, L. (1995). *ST : A Computer Program for IRT Scale Transformation*.