More on the Computation of Higher-Order Derivatives of the Elementary Symmetric Functions in the Rasch Model

Michelle Liou

Academia Sinica

A recursive equation for computing higher-order derivatives of the elementary symmetric functions in the Rasch model is proposed. The formula is conceptually simple and relatively more efficient than the sum algorithm (Gustafsson, 1980). A simulation study indicated that the proposed formula has a small loss in accuracy, compared to the sum algorithm, for computing higher-order derivatives when tests contained 60 items or less. *Index terms: difference algorithm, elementary symmetric functions, Formann's equation, Jansen's equation, Rasch model, sum algorithm.*

The Rasch model posits that the distribution of a random observation u_i, representing a 0 or 1 response to item i, has the form:

$$P_i(u_i = 1 | \theta) = \frac{1}{1 + \exp(b_i - \theta)} , \qquad (1)$$

where θ is the trait level parameter, and b_i is the item difficulty parameter. In conditional maximum likelihood estimation of the b_i (Gustafsson, 1980) as well as in calculating sampling distributions of many person-fit statistics for assessing the consistency of response patterns to the Rasch model (Glas, 1988; Klauer, 1991; Molenaar & Hoijtink, 1990), the following elementary symmetric functions (ESFs) must be evaluated for an n-item test:

$$\begin{aligned}
\gamma_j &= 0 (j < 0)\\
\gamma_0 &= 1\\
\gamma_1 &= \varepsilon_1 + \ldots + \varepsilon_n\\
\gamma_2 &= \varepsilon_1\varepsilon_2 + \varepsilon_1\varepsilon_3 + \ldots + \varepsilon_{i-1}\varepsilon_i + \ldots + \varepsilon_{n-1}\varepsilon_n\\
&\vdots\\
\gamma_n &= \varepsilon_1\varepsilon_2 \ldots \varepsilon_n ,
\end{aligned} \qquad (2)$$

where $\varepsilon_i = \exp(-b_i)$ for $i = 1, \ldots, n$, and is collected in vector $\boldsymbol{\varepsilon}$. γ_q denotes the symmetric function of order q, and $\gamma_q = 0$ for $q > n$.

In addition to γ_q for $q = 0, 1, \ldots, n$, many person-fit statistics also require the calculation of higher-order derivatives of the ESFs in Equation 2. For example, the central moments of the M statistic (Molenaar & Hoijtink, 1990, p. 96) require the first-, second-, and third-order derivatives of the ESFs $\partial\gamma_q/\partial\varepsilon_i$, $\partial\gamma_q/\partial\varepsilon_i\partial\varepsilon_j$, and $\partial\gamma_q/\partial\varepsilon_i\partial\varepsilon_j\partial\varepsilon_k$, respectively, for $\varepsilon_i \neq \varepsilon_j \neq \varepsilon_k$; the R_{2m} and R_{2c} statistics (Glas, 1988, p. 538) require the fourth-order derivatives of ESFs $\partial\gamma_q/\partial\varepsilon_i\partial\varepsilon_j\partial\varepsilon_k\partial\varepsilon_s$ for $\varepsilon_i \neq \varepsilon_j \neq \varepsilon_k \neq \varepsilon_s$. For simplification, the higher-order derivatives of the ESFs are denoted as $\gamma^{(i)}_{q-1}$, $\gamma^{(i,j)}_{q-2}$, $\gamma^{(i,j,k)}_{q-3}$, $\gamma^{(i,j,k,s)}_{q-4}$, and

APPLIED PSYCHOLOGICAL MEASUREMENT
Vol. 18, No. 1, March 1994, pp. 53-62
© Copyright 1994 Applied Psychological Measurement Inc.
0146-6216/94/010053-10$1.75

second-order derivatives $\gamma_{q-2}^{(i,j)}$ for $q = 1, \ldots, n$ are determined by eliminating ε_i and ε_j and computing the ESFs as in Equation 2 for the rest of the elements in ε. Analogously, higher-order derivatives can be obtained by eliminating a greater number of parameters.

Two algorithms (Fischer, 1974)—the difference algorithm and the sum algorithm (Gustafsson, 1980)—have been used for computing ESFs. The difference algorithm computes γ_q for $q = 1, \ldots, n$ using a recursive equation involving a subtraction operation that accumulates errors in rounding—especially when the ε_i values are spread over a wide range (Verhelst, Glas, & van der Sluis, 1984). On the other hand, the recursive equation in the sum algorithm involves only addition and multiplication operations; therefore, the main source of numerical instability is removed. However, the computational cost of the sum algorithm is higher than that for the difference algorithm. For mth-order derivatives, the sum algorithm must be run $\binom{n}{m}$ times—each time with different $(n - m)$ elements in ε. When n becomes large, a full implementation of the sum algorithm may not be feasible without using a high-speed computer.

An extension of the difference algorithm for computing higher-order derivatives of ESFs is proposed; it is conceptually simple for practitioners and computationally more efficient than the sum algorithm. A simulation study was conducted to evaluate the accuracy of this algorithm in comparison to the sum algorithm for test lengths of 60 items or less.

Computational Algorithms

The Sum Algorithm

The sum algorithm uses the following recursive equation:

$$\gamma_q(\varepsilon_1, \ldots, \varepsilon_i) = \varepsilon_i \gamma_{q-1}(\varepsilon_1, \ldots, \varepsilon_{i-1}) + \gamma_q(\varepsilon_1, \ldots, \varepsilon_{i-1}) \ , \tag{3}$$

where $i = 2, \ldots, n$ and $q = 1, \ldots, i$. Using the definitions $\gamma_0(\varepsilon_1) = 0$ and $\gamma_1(\varepsilon_1) = \varepsilon_1$, the sum algorithm computes γ_q for $i = 2$ and $q = 1, 2$, and so forth, until $i = n$ and $q = 1, \ldots, n$. Because it uses only addition and multiplication, Equation 3 is numerically stable. For evaluating the ESFs in Equation 2, the sum algorithm needs $n(n - 1)/2$ multiplications (not counting multiplications for γ_0 and γ_1) for n-item tests. Equation 3 also can be used to compute higher-order derivatives of the ESFs. For example, $\gamma_{q-2}^{(i,j)}$ can be found by eliminating ε_i and ε_j from vector ε and computing γ_q (using Equation 3) for the rest of the $n - 2$ elements in ε. For mth-order derivatives, therefore, the sum algorithm needs $\binom{n}{m}(n - m)(n - m - 1)/2$ multiplications—a function of n^{m+2}. For tests of more than 40 items, the algorithm may become increasingly inefficient for computing higher-order derivatives.

The Difference Algorithm

The γ_q values for $q = 0, \ldots, n$ and the related first-order derivatives can be computed simultaneously using the following equations (Gustafsson, 1980):

$$\gamma_q^{(i)} = \gamma_q - \varepsilon_i \gamma_{q-1}^{(i)} \ , \tag{4}$$

and

$$\gamma_q = 1/q \left(\sum_{i=1}^{n} \varepsilon_i \gamma_{q-1}^{(i)} \right) \ . \tag{5}$$

Starting with $\gamma_1 = \Sigma_i^n \varepsilon_i$ and $\gamma_0^{(i)} = 1$ for $i = 1, \ldots, n$, the difference algorithm uses Equations 4 and 5 alternately to provide a relatively fast computation of γ_q and $\gamma_{q-1}^{(i)}$ simultaneously. However, due to the respective subtraction and division operations in Equations 4 and 5, the algorithm accumulates rounding error and becomes increasingly less accurate as n increases. In order to avoid rounding errors while preserving computational efficiency, Gustafsson (1980) suggested computing γ_qs that are numerically less stable with the sum algorithm, and solving for $\gamma_{q-1}^{(i)}$ recursively with Equation 4.

Before proposing an extension of the difference algorithm for computing higher-order derivatives, the union property of ESFs (Verhelst, Glas, & van der Sluis, 1984) is stated as a lemma.

Lemma 1. For an arbitrary partitioning of n items into two groups, each containing n_1 and $n - n_1$ items, the ESFs resulting from the union of two groups are:

$$\gamma_q(\varepsilon_1, \ldots, \varepsilon_n) = \gamma_q[(\varepsilon_1, \ldots, \varepsilon_{n_1}) \cup (\varepsilon_{n_1+1}, \ldots, \varepsilon_n)] = \sum_{p=0}^{k} \gamma_p(\varepsilon_1, \ldots, \varepsilon_{n_1})\, \gamma_{q-p}(\varepsilon_{n_1+1}, \ldots, \varepsilon_n) \ , \tag{6}$$

where $k = \text{Min}(n_1, n - n_1)$ and $\gamma_p(\varepsilon_{n_1+1}, \ldots, \varepsilon_n) = 0$ for $p > n - n_1$. This lemma is proven using both the general distributive law and the algebraic independence of ESFs [see MacLane & Birkhoff (1988) for a discussion of the algebraic properties of ESFs].

Lemma 1 was used to derive the recursive equation for computing higher-order derivatives. Take the element ε_i from vector ε in the first group (i.e., $n_1 = 1$) and leave the rest of the elements in ε in the second group. For simplification, let $\tilde{\gamma}_0 = 1$ and $\tilde{\gamma}_1 = \varepsilon_i$ be the ESFs in the first group, and $\gamma_0^{(i)}, \ldots,$ and $\gamma_{n-1}^{(i)}$ be the ESFs in the second group. Based on Lemma 1, the ESFs of the n elements become:

$$\begin{aligned}
\gamma_0 &= 1 \ , \\
\gamma_1 &= \tilde{\gamma}_0\gamma_1^{(i)} + \tilde{\gamma}_1\gamma_0^{(i)} = \gamma_1^{(i)} + \varepsilon_i \ , \\
\gamma_2 &= \gamma_2^{(i)} + \varepsilon_i\gamma_1^{(i)} \ , \\
&\vdots \\
\gamma_n &= \varepsilon_i\gamma_{n-1}^{(i)} \ .
\end{aligned} \tag{7}$$

For numerical accuracy, the γ_q values on the left side of Equation 7 can be determined by the sum algorithm (Gustafsson, 1980). Thus, the $\gamma_q^{(i)}$ values can be solved successively using the equality

$$\gamma_q^{(i)} = \gamma_q - \varepsilon_i\gamma_{q-1}^{(i)} \tag{8}$$

for $q = 1, \ldots, n - 1$. The recursive equation starts with $\gamma_0^{(i)} = 1$ and is equivalent to Equation 4 in the difference algorithm.

The Extended Algorithm

Based on Lemma 1, the recursive equation for higher-order derivatives can be derived analogously. For example, the third-order derivatives can be derived by taking the elements ε_i, ε_j, and ε_k from vector ε as the first group (i.e., $n_1 = 3$), and leaving the rest of the elements in ε in the second group. Let

$$\tilde{\gamma}_0 = 1 \ , \tag{9}$$

$$\tilde{\gamma}_1 = \varepsilon_i + \varepsilon_j + \varepsilon_k \ , \tag{10}$$

$$\tilde{\gamma}_2 = \varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_j\varepsilon_k \ , \tag{11}$$

and

$$\tilde{\gamma}_3 = \varepsilon_i\varepsilon_j\varepsilon_k \tag{12}$$

denote the ESFs in the first group and $\gamma_0^{(i,j,k)}, \ldots, \gamma_{n-3}^{(i,j,k)}$ be the ESFs in the second group. According to Lemma 1, the ESFs of the n elements become:

$$\begin{aligned}
\gamma_0 &= 1 \\
\gamma_1 &= \tilde{\gamma}_0\gamma_1^{(i,j,k)} + \tilde{\gamma}_1\gamma_0^{(i,j,k)} = \gamma_1^{(i,j,k)} + (\varepsilon_i + \varepsilon_j + \varepsilon_k), \\
\gamma_2 &= \gamma_2^{(i,j,k)} + (\varepsilon_i + \varepsilon_j + \varepsilon_k)\gamma_1^{(i,j,k)} + (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_j\varepsilon_k), \\
&\vdots \\
\gamma_n &= \varepsilon_i\varepsilon_j\varepsilon_k\gamma_{n-3}^{(i,j,k)} \ .
\end{aligned} \tag{13}$$

After the γ_q values have been determined by the sum algorithm, $\gamma_{q-3}^{(i,j,k)}$ can be solved for the recursive equation

$$\gamma_q^{(i,j,k)} = \gamma_q - [(\varepsilon_i + \varepsilon_j + \varepsilon_k)\gamma_{q-1}^{(i,j,k)} + (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_j\varepsilon_k)\gamma_{q-2}^{(i,j,k)} + \varepsilon_i\varepsilon_j\varepsilon_k \ \ \gamma_{q-3}^{(i,j,k)}] \ . \tag{14}$$

Starting with the value $\gamma_0^{(i,j,k)} = 1$, the algorithm successively solves $\gamma_q^{(i,j,k)}$ for $q = 1, \ldots, n - 3$.

The subtraction operation in Equation 14 also leads to an accumulation of rounding error. In order to improve the accuracy of the algorithm, the $\gamma_q^{(i,j,k)}$ functions can be evaluated using Equation 14 for $q = 1, \ldots, (n - 3)/2$. The remaining ESFs then can be evaluated using the backward algorithm

$$\gamma_q^{(i,j,k)} = [\gamma_{q+3} - \gamma_{q+3}^{(i,j,k)} - (\varepsilon_i + \varepsilon_j + \varepsilon_k)\gamma_{q+2}^{(i,j,k)} - (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_j\varepsilon_k)\gamma_{q+1}{}^{(i,j,k)}]/\varepsilon_i\varepsilon_j\varepsilon_k \ . \tag{15}$$

The backward algorithm starts with $\gamma_{n-3}^{(i,j,k)} = \gamma_n/\varepsilon_i\varepsilon_j\varepsilon_k$ and successively solves for the rest of the third-order derivatives. A summary of the forward and backward equations for computing the first- through fourth-order derivatives of ESFs with the extended algorithm is presented in Table 1. For mth-order derivatives, the total number of multiplications required for the extended algorithm is approximately $n(n + 1)/2 + \binom{n}{m}(n - m)[\sum_i^{m-1}(i - 1)\binom{m}{i} + 4]$, which is a function of n^{m+1}. Therefore, the computational cost of the extended algorithm is approximately $1/n$th the cost of the sum algorithm.

Jansen's Equation

Jansen (1984) also proposed a method for computing $\gamma_{q-2}^{(i,j)}$, which is a simple modification of the difference algorithm and can be extended easily to obtain higher-order derivatives. Second-order

Table 1
The Forward and Backward Equations for Computing the Derivatives of the Elementary Symmetric Functions in the Extended Algorithm

Derivatives	Forward Equation	Backward Equation
First-Order	$\gamma_q^{(i)} = \gamma_q - \varepsilon_i\gamma_{q-1}^{(i)}$	$\gamma_q^{(i)} = [\gamma_{q+1} - \gamma_{q+1}^{(i)}]/\varepsilon_i$
Second-Order	$\gamma_q^{(i,j)} = \gamma_q - [(\varepsilon_i + \varepsilon_j)\gamma_{q-1}^{(i,j)} + \varepsilon_i\varepsilon_j\gamma_{q-2}^{(i,j)}]$	$\gamma_q^{(i,j)} = \{\gamma_{q+2} - [\gamma_{q+2}^{(i,j)} + (\varepsilon_i + \varepsilon_j)\gamma_{q+1}^{(i,j)}]\}/\varepsilon_i\varepsilon_j$
Third-Order	$\gamma_q^{(i,j,k)} = \gamma_q - [(\varepsilon_i + \varepsilon_j + \varepsilon_k)\gamma_{q-1}^{(i,j,k)}$ $+ (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_j\varepsilon_k)\gamma_{q-2}^{(i,j,k)}$ $+ \varepsilon_i\varepsilon_j\varepsilon_k\gamma_{q-3}^{(i,j,k)}]$	$\gamma_q^{(i,j,k)} = \{\gamma_{q+3} - [\gamma_{q+3}^{(i,j,k)} + (\varepsilon_i + \varepsilon_j + \varepsilon_k)\gamma_{q+2}^{(i,j,k)}$ $+ (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_j\varepsilon_k)\gamma_{q+1}^{(i,j,k)}]\}/\varepsilon_i\varepsilon_j\varepsilon_k$
Fourth-Order	$\gamma_q^{(i,j,k,s)} = \gamma_q - [(\varepsilon_i + \varepsilon_j + \varepsilon_k + \varepsilon_s)\gamma_{q-1}^{(i,j,k,s)}$ $+ (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_i\varepsilon_s + \varepsilon_j\varepsilon_k + \varepsilon_j\varepsilon_s$ $+ \varepsilon_k\varepsilon_s)\gamma_{q-2}^{(i,j,k,s)} + (\varepsilon_i\varepsilon_j\varepsilon_k + \varepsilon_i\varepsilon_j\varepsilon_s + \varepsilon_i\varepsilon_k\varepsilon_s$ $+ \varepsilon_j\varepsilon_k\varepsilon_s)\gamma_{q-3}^{(i,j,k,s)} + \varepsilon_i\varepsilon_j\varepsilon_k\varepsilon_s\gamma_{q-4}^{(i,j,k,s)}]$	$\gamma_q^{(i,j,k,s)} = \{\gamma_{q+4} - [\gamma_{q+4}^{(i,j,k,s)} + (\varepsilon_i + \varepsilon_j + \varepsilon_k$ $+ \varepsilon_s)\gamma_{q+3}^{(i,j,k,s)} + (\varepsilon_i\varepsilon_j + \varepsilon_i\varepsilon_k + \varepsilon_i\varepsilon_s$ $+ \varepsilon_j\varepsilon_k + \varepsilon_j\varepsilon_s\varepsilon_k\varepsilon_s)\gamma_{q+2}^{(i,j,k,s)} + (\varepsilon_i\varepsilon_j\varepsilon_k + \varepsilon_i\varepsilon_j\varepsilon_s$ $+ \varepsilon_i\varepsilon_k\varepsilon_s + \varepsilon_j\varepsilon_k\varepsilon_s)\gamma_{q+1}^{(i,j,k,s)}]\}/\varepsilon_i\varepsilon_j\varepsilon_k\varepsilon_s$

derivatives of the ESFs can be derived using the results obtained from Equation 4. Jansen (1984) proved that if $\varepsilon_i \neq \varepsilon_j$, then

$$\gamma^{(i,j)}_{q-2} = [\gamma^{(i)}_{q-1} - \gamma^{(j)}_{q-1}]/[\varepsilon_j - \varepsilon_i] \ . \qquad (16)$$

Higher-order derivatives also can be determined as analogs of Equation 16. For example, third-order derivatives can be obtained by first computing $\gamma^{(i,k)}_{q-2}$ and $\gamma^{(j,k)}_{q-2}$ with Equations 4 and 16, then computing

$$\gamma^{(i,j,k)}_{q-3} = [\gamma^{(i,k)}_{q-2} - \gamma^{(j,k)}_{q-2}]/[\varepsilon_j - \varepsilon_i] \ . \qquad (17)$$

Equation 16 requires subtraction and division and, therefore, is less accurate than the recursive equation in the sum algorithm.

Accuracy of Jansen's equation. The accuracy of Jansen's equation (Equation 16) is affected by the relative sizes of ε_i and ε_j. For example, consider an 11-item test with *b*s of -3.00000001, -3.0, -2.9999999, -2.0, -.5, 0.0, .00000001, 1.0, 2.0, 2.1, and 3.0. The $\gamma^{(i,j)}_{q-2}$ values for $q = 1, \ldots, 9$ were computed independently using the sum algorithm, the extended algorithm, and Jansen's equation. The $\gamma^{(i)}_q$ values in Jansen's equation were computed using the forward and backward equations in Table 1, and the $\gamma^{(i,j)}_q$ values then were obtained using Equation 16. The loss in accuracy due to Jansen's equation compared to the sum algorithm was measured by the relative error

$$RE^{(i,j)}_q = [\,|S^{(i,j)}_q - J^{(i,j)}_q|\,/S^{(i,j)}_q] \ , \qquad (18)$$

where $S^{(i,j)}_q$ denotes the ESFs of order q computed by the sum algorithm, and $J^{(i,j)}_q$ denotes the same function computed by Jansen's equation. The first column of Table 2 gives the average $RE^{(i,j)}_q$ over the $\binom{n}{2}$ runs of Jansen's equation for $q = 1, \ldots, 9$. The average relative errors of the extended algorithm compared to the sum algorithm were computed as analogs of Equation 18 and are shown as Example 1 in Table 2. For this example, Jansen's equation yielded much less accurate results than those for the extended algorithm.

Numerical problems with Jansen's equation. Jansen's equation is also subject to numerical problems when $\varepsilon_i = \varepsilon_j$. Formann (1986) proposed a method for dealing with these situations: Apply Equation 4 to $\gamma^{(i)}_{q-1}$ with respect to each item i and obtain

$$\sum_{\substack{j=1 \\ i\neq j}}^{n} \varepsilon_j \gamma^{(i,j)}_{q-2} = (q-1)\gamma^{(i)}_{q-1} \ . \qquad (19)$$

Table 2
The Average Relative Errors of Different Algorithms
Compared to the Sum Algorithm for Computing
Second-Order Derivatives in Two Examples

	Example 1		Example 2	
q	Jansen	Extended	Formann	Extended
1	6.688E–09	3.510E–18	4.832E–16	1.118E–17
2	1.446E–09	3.321E–17	1.627E–16	2.231E–17
3	1.360E–09	1.143E–16	4.655E–16	1.158E–16
4	2.982E–10	2.091E–15	2.949E–15	1.185E–15
5	1.732E–08	3.171E–15	1.565E–14	6.517E–16
6	1.217E–09	1.845E–16	4.049E–17	6.640E–17
7	1.572E–10	5.681E–17	2.278E–17	2.341E–17
8	7.398E–11	2.668E–17	2.397E–17	1.928E–17
9	3.268E–11	2.130E–17	2.102E–17	8.513E–18

Table 3
CPU Time in Seconds Needed on VAX and SUN Computers for Evaluation of Higher-Order Derivatives of the Elementary Symmetric Functions for the Extended (Ext), Sum, and Jansen Formulas

b Distribu-		First-Order		Second-Order			Third-Order			Fourth-Order		
tion and *n*		Ext	Sum	Jansen	Ext	Sum	Jansen	Ext	Sum	Jansen	Ext	Sum
U(−1.5, 1.5)												
20	Vax	.17	.78	.12	.10	7.37	.46	.43	44.37	1.77	2.01	186.20
	SUN	0.00	.05	.01	.01	.41	.03	.01	2.46	.12	.07	10.66
40	Vax	.07	1.98	.63	.53	39.05	7.38	7.74	487.73	65.93	87.76	4,439.81
	SUN	.01	.10	.05	.02	1.98	.55	.27	24.94	5.13	2.90	227.83
60	Vax	.10	4.04	2.07	1.73	120.15	41.81	41.62	2,282.78	551.19	735.11	32,012.95
	SUN	0.00	.18	.15	.07	5.47	2.93	1.48	104.67	40.89	24.51	1,477.14
U(−2.5, 2.5)												
20	Vax	.20	.77	.12	.10	7.33	.49	.50	44.52	1.81	2.21	184.52
	SUN	0.00	.05	.01	.01	.42	.04	.02	2.52	.13	.07	10.43
40	Vax	.07	2.11	.66	.52	38.49	7.99	8.92	486.55	79.70	91.39	4,426.59
	SUN	0.00	.10	.05	.02	1.97	.55	.28	24.79	4.96	2.89	227.19
60	Vax	.10	4.26	2.14	1.73	119.64	42.03	42.89	2,292.51	648.60	758.94	32,318.80
	SUN	0.00	.19	.16	.07	5.48	2.92	1.48	104.56	40.87	24.47	1,474.98
N(−1.5, 1.5)												
20	Vax	.16	.77	.12	.09	7.23	.49	.47	44.55	1.71	1.98	186.14
	SUN	0.00	.05	.01	.01	.41	.03	.01	2.47	.12	.07	10.55
40	Vax	.07	1.97	.65	.53	38.44	8.44	8.39	485.52	74.31	90.77	4,419.43
	SUN	0.00	.11	.05	.02	2.01	.56	.28	24.83	4.96	2.94	227.69
60	Vax	.11	4.18	2.25	1.73	118.75	41.77	44.50	2,289.44	616.44	804.34	32,322.59
	SUN	.01	.19	.16	.06	5.47	2.93	1.48	104.65	40.88	24.49	1,476.96
N(−2.5, 2.5)												
20	Vax	.18	.77	.11	.14	7.22	.44	.44	43.98	1.74	2.03	183.16
	SUN	0.00	.05	.01	0.00	.41	.03	.02	2.48	.12	.07	10.52
40	Vax	.07	1.96	.63	.65	37.78	7.61	8.18	483.10	74.36	87.10	4,406.42
	SUN	.01	.10	.05	.02	2.03	.56	.28	24.80	5.15	2.90	227.80
60	Vax	.12	4.21	2.15	1.99	121.92	40.26	46.36	2,305.93	631.77	775.93	32,204.46
	SUN	0.00	.19	.15	.07	5.47	2.94	1.48	104.69	40.90	24.52	1,477.78

Because all $\gamma_{q-2}^{(i,j)}$ values can be computed according to Equation 16 when $\varepsilon_i \neq \varepsilon_j$, the sum $G^{(i)}$ for all remaining $\gamma_{q-2}^{(i,j)}$ values in which $\varepsilon_i = \varepsilon_j$ equals

$$G^{(i)} = \sum_{\substack{j=1 \\ i \neq j, \varepsilon_i = \varepsilon_j}}^{n} \varepsilon_j \gamma_{q-2}^{(i,j)} = (q-1)\gamma_{q-1}^{(i)} - \sum_{\substack{j=1 \\ i \neq j, \varepsilon_i \neq \varepsilon_j}}^{n} \varepsilon_j \gamma_{q-2}^{(i,j)} \; . \tag{20}$$

If k of the $n - 1$ components in ε are identical in size to ε_i, then $\gamma_{q-2}^{(i,j)} = G^{(i)}/k\varepsilon_j$ for $\varepsilon_i = \varepsilon_j$. The accuracy of $G^{(i)}$ depends on the accuracy of $\gamma_{q-1}^{(i)}$, but not on the parameters ε_i and ε_j or their difference. For example, consider an 11-item test with *b*s of -3.0, -3.0, -2.0, -1.0, -1.0, -1.0, 0.0, 1.0, 1.5, 2.0, and 2.0. Using Formann's equation and the extended algorithm, the $\gamma_q^{(i,j)}$ values were computed for the 11 parameters. The relative errors of Formann's equation and the extended algorithm compared to the sum algorithm are presented as Example 2 in Table 2. For this example, Formann's equation yielded results as accurate as those from the extended algorithm.

Method

Simulated Datasets

Item difficulties (b parameters) in the Rasch model were generated randomly from uniform and standard normal distributions. Four distributions [U(-1.5,1.5), U(-2.5,2.5), N(-1.5,1.5), and N(-2.5,2.5)] of b were used with tests of $n = 20$, 40, or 60 items, resulting in 12 combinations of distribution and n (see Table 3). Random samples were generated separately for the three test lengths. The ε_i parameters for $i = 1, \ldots, n$ were computed based on the random b values, that is, $\varepsilon_i = \exp(-b_i)$.

Procedure

Because the simulated parameters had distinct values, the first- through fourth-order derivatives were computed for different test lengths using the sum algorithm, the extended algorithm, and Jansen's equation. The higher-order derivatives were computed independently by implementing the three algorithms on a VAX-8350 computer and a SUN SPARC-10 workstation using Fortran 77 programs. All program variables were declared to be of double precision (REAL*8). In order to improve the accuracy of Jansen's equation, $\gamma_q^{(i)}$ was computed using the forward and backward equations in Table 1, and γ_q values were evaluated using the sum algorithm. Therefore, the extended algorithm and Jansen's equation yielded identical results for computing the first-order derivatives.

The losses in accuracy using Jansen's equation and the extended algorithm compared to the sum algorithm were measured by *RE* (Equation 18). Let $RE_q^{(m)}$ denote the average relative error over the $\binom{n}{m}$ runs of an algorithm for the mth-order derivatives of the ESFs of order q. For ease of comparison, the maximum, minimum, and average values of $RE_q^{(m)}$ for $q = 1, \ldots, n - m$ were computed.

Results

Table 3 shows the computing time required for computing higher-order derivatives using different algorithms and computers. Both Jansen's equation and the extended algorithm were faster than the sum algorithm. For example, for 60-item tests the sum algorithm took more than 30 minutes [2,282.78 seconds for U(-1.5,1.5)] to compute third-order derivatives and more than eight hours to compute fourth-order derivatives on the VAX-8350. For the same situation, the extended algorithm required less than one minute to compute third-order derivatives and less than 13 minutes to compute fourth-order derivatives. The extended algorithm was slightly more efficient than Jansen's equation for computing second- and third-order derivatives, but was slower than Jansen's equation for computing

Table 4
Average, Maximum, and Minimum of Relative Errors Compared to the Sum Algorithm of the Jansen and Extended Algorithms for First- Through Fourth-Order Derivatives

b Distribution,		First-Order	Second-Order		Third-Order		Fourth-Order	
n, and Statistic		Extended	Jansen	Extended	Jansen	Extended	Jansen	Extended
U(−1.5, 1.5)								
20	Average	1.776E−17	1.665E−16	4.874E−17	1.501E−16	8.854E−17	8.874E−16	2.488E−16
	Maximum	3.674E−17	3.773E−16	3.199E−16	4.308E−16	5.169E−16	6.529E−15	2.391E−15
	Minimum	4.930E−18	5.270E−17	6.559E−18	4.370E−17	9.917E−18	5.980E−17	9.062E−18
40	Average	8.442E−16	2.829E−15	2.346E−15	1.494E−14	1.336E−14	4.654E−14	2.766E−14
	Maximum	2.266E−14	5.437E−14	4.868E−14	3.851E−13	3.254E−13	6.591E−13	3.838E−13
	Minimum	2.986E−18	4.122E−17	6.589E−18	4.609E−17	1.042E−17	6.584E−17	7.717E−18
60	Average	3.014E−14	5.500E−13	5.804E−13	1.120E−12	1.119E−12	1.430E−11	8.407E−12
	Maximum	1.282E−12	2.273E−11	2.588E−11	4.468E−11	4.704E−11	6.059E−10	3.687E−10
	Minimum	1.129E−17	6.586E−17	1.826E−17	1.028E−16	2.080E−17	1.375E−16	2.022E−17
U(−2.5, 2.5)								
20	Average	1.195E−15	2.763E−15	3.226E−15	2.929E−14	1.642E−14	1.991E−14	1.136E−14
	Maximum	1.917E−14	3.612E−14	4.319E−14	4.484E−13	2.462E−13	1.892E−13	1.140E−13
	Minimum	3.892E−18	2.720E−17	4.918E−18	3.117E−17	1.202E−17	3.594E−17	1.996E−17
40	Average	6.388E−11	1.047E−09	1.049E−09	1.042E−09	1.143E−09	2.583E−08	8.943E−09
	Maximum	2.259E−09	3.682E−08	3.685E−08	3.541E−08	3.895E−08	8.625E−07	3.007E−07
	Minimum	1.018E−17	4.290E−17	1.645E−17	4.583E−17	1.939E−17	5.718E−17	1.753E−17
60	Average	2.975E−05*	5.211E−04*	5.333E−04*	2.930E−04*	2.933E−04*	7.801E−03*	3.307E−03*
	Maximum	1.643E−03*	2.845E−02*	2.912E−02*	1.570E−02*	1.572E−02*	.4123*	.1750*
	Minimum	2.505E−18	4.896E−17	6.549E−18	5.485E−17	8.173E−18	7.338E−17	1.071E−17
N(−1.5, 1.5)								
20	Average	1.982E−17	1.187E−16	6.890E−17	1.971E−16	1.447E−16	1.495E−15	3.653E−16
	Maximum	5.889E−17	3.999E−16	4.064E−16	9.363E−16	9.089E−16	1.306E−14	3.154E−15
	Minimum	7.645E−18	2.698E−17	6.586E−18	2.999E−17	6.051E−18	4.436E−17	7.681E−18
40	Average	2.391E−16	2.559E−15	1.633E−15	3.259E−15	2.930E−15	4.718E−14	1.239E−14
	Maximum	5.238E−15	6.143E−14	4.019E−14	5.648E−14	5.567E−14	1.053E−12	2.995E−13
	Minimum	2.490E−18	8.809E−17	7.319E−18	7.157E−17	7.181E−18	1.380E−16	9.856E−18
60	Average	5.368E−15	5.728E−12	1.427E−13	1.379E−12	2.966E−13	7.811E−12	2.101E−12
	Maximum	1.959E−13	2.557E−10	6.046E−12	5.807E−11	1.030E−11	3.334E−10	8.719E−11
	Minimum	9.120E−18	6.511E−17	2.042E−17	9.863E−17	2.073E−17	1.210E−16	2.103E−17
N(−2.5, 2.5)								
20	Average	7.046E−17	5.441E−16	1.503E−16	9.524E−16	6.826E−16	2.958E−15	7.006E−16
	Maximum	7.197E−16	3.869E−15	1.431E−15	8.992E−15	8.474E−15	2.301E−14	5.969E−15
	Minimum	5.815E−18	5.316E−17	1.013E−17	5.843E−17	1.196E−17	1.019E−16	1.115E−17
40	Average	1.105E−12	6.679E−12	7.463E−12	3.144E−12	4.958E−12	1.536E−10	2.402E−11
	Maximum	3.782E−11	2.263E−10	2.530E−10	1.006E−10	1.611E−10	4.910E−10	7.730E−10
	Minimum	5.894E−18	6.435E−17	1.296E−17	1.400E−16	1.556E−17	1.592E−16	1.631E−17
60	Average	3.121E−09	1.466E−08	8.507E−09	1.820E−08	1.100E−08	7.158E−07	2.544E−08
	Maximum	9.481E−08	5.679E−07	3.959E−07	6.258E−07	3.145E−07	3.431E−05*	1.057E−06
	Minimum	6.220E−18	9.040E−17	7.164E−18	9.518E−17	1.376E−17	1.561E−16	1.575E−17

*$RE > 10^{-5}$.

fourth-order derivatives on the VAX-8350. However, the extended algorithm was faster than Jansen's equation on the SUN SPARC-10 for the second- through fourth-order derivatives. The computing time required for implementing the algorithms on the SUN SPARC-10 was approximately 1/20th of that required using the VAX-8350. Therefore, use of the extended algorithm and Jansen's equation becomes less significant on a high-speed computer. The speed of the VAX-8350 is comparable to that of a DOS-based 386 personal computer. The computing times listed in Table 3, therefore, also indicate the approximate computing times for the algorithms on such a computer.

Table 4 presents the maximum, minimum, and average $RE_q^{(m)}$ values over the $n - m$ values of q for $m = 1, \ldots, 4$ on the VAX-8350. When $n = 60$ and the b parameters were uniformly distributed over a wide range (i.e., -2.5, 2.5), both Jansen's equation and the extended algorithm yielded average *RE*s that were greater than 10^{-5} (marked with a * in Table 4). In addition, both algorithms were highly inaccurate when the *b*s were normally distributed over a wide range. Therefore, the range of the *b*s played an important role in the computational accuracy of the two algorithms. However, in general the extended algorithm was slightly more accurate than Jansen's equation. The simulation results from the use of the SUN SPARC-10 differed in no important way from the results listed in Table 4.

Discussion

In practice, higher-order derivatives of ESFs often are evaluated repeatedly. For example, person-fit statistics in the Rasch model are computed for numerous response patterns simultaneously in a large dataset, and likelihood equations are evaluated iteratively in conditional maximum likelihood estimation of the b parameters. Computational efficiency in these cases can become an overriding consideration. Simulation results suggested that the extended algorithm is a fast alternative to the sum algorithm for evaluating higher-order derivatives of ESFs when tests contain a large number of items and the b parameters are within a small range (e.g., -1.5 to 1.5). Admittedly, the sum algorithm may be used to obtain more accurate results for short tests as well as for longer tests if a high-speed computer is available. On the other hand, the accuracy of the extended algorithm can be improved further by requiring a higher level of computational precision (REAL*16) for program variables. However, this may cause a loss in efficiency. For example, with more precise computation, the extended algorithm gave an average $RE_q^{(3)}$ of 4.217E-24 in 12 minutes, and an average $RE_q^{(4)}$ of 3.248E-23 in 3.5 hours on a VAX-8350 for the 60-item test for U(-2.5,2.5). The loss in accuracy of the extended algorithm was larger when the sizes of the eliminated item parameters were quite different and was smaller otherwise. Accordingly, researchers may selectively use more precise computation for derivatives that contain larger errors when the b parameters spread over a wide range.

References

Fischer, G. H. (1974). *Einfuhrung in Die Theorie Psychologischer Tests* [Introduction to the theory of psychological tests]. Bern: Verlag Hans Huber.

Formann, A. K. (1986). A note on the computation of the second-order derivatives of the elementary symmetric functions in the Rasch model. *Psychometrika, 51,* 335–339.

Glas, C. A. (1988). The derivation of some tests for the Rasch model from the multinomial distribution. *Psychometrika, 53,* 525–546.

Gustafsson, J. E. (1980). A solution of the conditional estimation problem for long tests in the Rasch model for dichotomous items. *Educational and Psychological Measurement, 40,* 377–385.

Jansen, P. G. (1984). Computing the second-order derivatives of the symmetric functions in the Rasch model. *Kwantitatieve Methoden, 13,* 131–147.

Klauer, K. C. (1991). An exact and optimal standardized person test for assessing consistency with the Rasch model. *Psychometrika, 56,* 213–228.

MacLane, S., & Birkhoff, G. (1988). *Algebra* (3rd ed.). New York: Chelsea.

Molenaar, I. W., & Hoijtink, H. (1990). The many null distributions of person fit indices. *Psychometrika, 55,* 75–106.

Verhelst, N. D., Glas, C. A., & van der Sluis, A. (1984).

Estimation problems in the Rasch model: The basic symmetric functions. *Computational Statistics Quarterly, 1,* 245–262.

Author's Address

Send requests for reprints or further information to Michelle Liou, Institute of Statistical Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. E-mail: stliou@twnas886.bitnet.