EPM: A Strategy for Principal Axis Minimization

Ali Ünlü and Michael D. Kickmeier-Rust

***Abstract*—This article introduces a three-round procedure, EPM (Extended Principal Axis Minimization), for automatic generation and testing of multiple sets of start values. EPM facilitates the use of Gegenfurtner's C implementation of Brent's PRincipal AXIS (PRAXIS) algorithm for function minimization without derivatives. It randomly generates vectors of start values for the model parameters, drawn from interactively specified, weighted intervals, and subsequently, performs iterative minimization loops based on best estimates resulting in the smallest minimum. Beside the PRAXIS algorithm's default stopping criterion on the difference between estimated parameter vectors, EPM applies an additional criterion on the difference between estimated minima. Moreover, the precision of Gegenfurtner's implementation can be improved using long double instead of double data types. EPM and PRAXIS are compared using three different functions.**

***Index Terms*—Automatic Generation of Start Values, Comparison of Implementations, Numerical Minimization, PRAXIS, Software, Usage Facilitation**

I. Introduction

Minimizing functions of several variables is a common and important problem in natural sciences. In this article, we propose an extension of Gegenfurtner's [1] C implementation of Brent's [2] PRincipal AXIS (PRAXIS) algorithm. PRAXIS is a frequently used algorithm for the numerical minimization of functions of several variables without the use of derivatives, and is a modification of a direction-set method by Powell [3]. For a brief overview of Powell's method and Brent's modification (PRAXIS), see [1]. Implementations, however, have some limitations. As other minimization routines, (a) PRAXIS converges to local minima, therefore, the results strongly depend on the selection of suitable start values, and (b) it's precision is limited by constraints like the applied stopping criterion and the implementation's data type used.

Ali Ünlü is with the Cognitive Science Section, Department of Psychology, University of Graz, Austria (phone: ++43-316-3805107; fax: ++43-316-3809806; e-mail: ali.uenlue@uni-graz.at).

Michael D. Kickmeier-Rust is with the Cognitive Science Section, Department of Psychology, University of Graz, Austria (e-mail: michael.kickmeier@uni-graz.at).

II. A Sketch of EPM

EPM (*Extended Principal Axis Minimization*) is a three-round strategy, extending the original implementation, which can be pursued for improved and effective applications of PRAXIS for optimization problems and to reduce the mentioned limitations. The EPM approach automatically generates and tests multiple sets of start values, randomly drawn from user specified weighted primary and secondary intervals (Round 1), and it refines and stabilizes the smallest Round 1 approximated minimum by means of certain iterative minimization loops and a stopping criterion on the change in approximated smallest minimum found in previous minimization runs and the minimum of the current run (Rounds 2 and 3). EPM is implemented as C (ANSI C 99) function. Additionally, to gain more precision for complex minimization problems, we converted the double data type used in the original PRAXIS implementation to the long double data type. The C implementation is held very general; so it might also be used with other minimization algorithms available in C/C++ without extensive modifications.

In Round 1, a number of start value vectors for the model parameters are randomly sampled from user specified weighted primary and secondary intervals (in the reals). The primary interval is determined by a center point C and a real number $d > 0$ as $[C - d, C + d]$. Two secondary intervals surrounding the primary interval are specified by a real number $e > d > 0$ as $[C - e, C - d)$ and $(C + d, C + e]$. Finally, the user specifies a weight $0 \leq w \leq 1$, meaning a start value is chosen randomly from the primary interval with probability w, and accordingly, with probability $1 - w$ from one of the secondary intervals. Subsequently, for each start value vector a minimization run is performed using the PRAXIS algorithm. The vector of parameter estimates resulting in the smallest minimum is passed to Round 2, for refinement and stabilization of the associated approximate minimum.

In Round 2, an iterative process approaches the true minimum. Starting with the start value vector passed by Round 1, EPM iteratively proceeds a number of minimization runs. In each run, the resulting vector of parameter estimates is used as new vector of start values for the following run. This procedure allows for a refined, closer approximation to the true minimum. Round 2 ends up with the smallest minimum of the minimization runs and the corresponding vector of parameter estimates. This vector is passed to Round 3 as new vector of start values.

In Round 3, the purpose is to stabilize the final minimum obtained in Round 2. The user specifies a maximum number of iterative minimization runs and a (small) difference criterion $c \geq 0$. Starting with the start value vector passed by Round 2, in each run, the resulting vector of parameter estimates is again used as new vector of start values for the following run. Whereas PRAXIS uses a stopping criterion on the difference between estimated parameter vectors, EPM applies a stopping criterion on the difference between the smallest minimum m_{glob} found in all previous minimization runs and the minimum m_{cur} of the current run. EPM stops the iterative loops in Round 3 when either the maximum number of iterations is reached or the criterion $|m_{\text{cur}} - m_{\text{glob}}| \leq c$ is fulfilled.

III. A Brief Comparison of EPM with the Original PRAXIS Implementation

To illustrate the benefits of EPM to the original PRAXIS implementation, we ran basic minimization trials using three different functions. We compared the minimization results based on various start values, PRAXIS settings, and double as well as long double versions.

A. Rosenbrock function

The Rosenbrock function is often used as a test problem for optimization algorithms. It is a two-parameter unimodal function ($x, y \in \mathbb{R}$)

$$f(x, y) := (1-x)^2 + 100 \cdot (y - x^2)^2,$$

which has a global minimum 0 at point (1, 1). A Mathematica plot of this function is depicted in Figure 1.

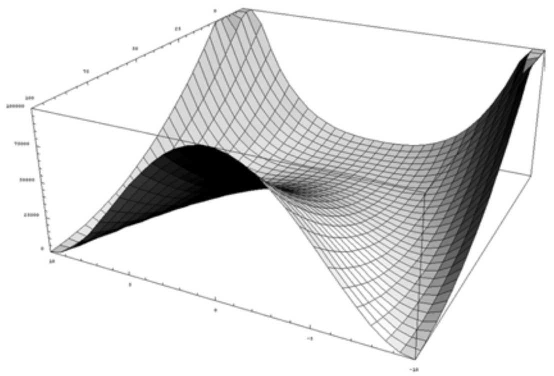

Fig. 1. Plot of the Rosenbrock function.

In Table I, we list the minimization results for this function.

Double versions: As expected, the selection of start values was crucial for the minimization results. We found minima ranging from 3.567E-03 to 5.265E+13 (*maxfun* = 1) and 1.173E-18 to 3.334E+03 (*maxfun* = 0) using the double version of the original PRAXIS implementation with different start values. The double version of EPM resulted in minima 3.420E-04 (*maxfun* = 1) and 1.171E-18 (*maxfun* = 0). The absolute values of the differences between the smallest approximated minima under PRAXIS and EPM were 3.225E-03 (*maxfun* = 1) and 1.537E-21 (*maxfun* = 0).

The Euclidean distances of the vectors of parameter estimates to the point (1, 1) at which the global minimum is attained ranged from 1.367E-01 to 3.241E+03 (*maxfun* = 1) and 2.423E-09 to 3.450E+03 (*maxfun* = 0) for the double version of PRAXIS. The double version of EPM resulted in Euclidean distances 4.099E-02 (*maxfun* = 1) and 2.420E-09 (*maxfun* = 0). The absolute values of the differences between the smallest Euclidean distances under PRAXIS and EPM were 9.576E-02 (*maxfun* = 1) and 3.507E-12 (*maxfun* = 0).

Long double versions: For the long double versions, we found minima ranging from 2.827E-01 to 2.235E+09 (*maxfun* = 1) and 3.899E-18 to 3.712E-14 (*maxfun* = 0) using PRAXIS, and minima 7.246E-12 (*maxfun* = 1) and 9.284E-23 (*maxfun* = 0) using EPM. The absolute values of the differences between the smallest approximated minima under PRAXIS and EPM were 2.827E-01 (*maxfun* = 1) and 3.899E-18 (*maxfun* = 0).

The Euclidean distances of the vectors of parameter estimates to the point (1, 1) ranged from 2.593E-01 to 6.984E+01 (*maxfun* = 1) and 1.406E-10 to 3.902E-08 (*maxfun* = 0) for the long double version of PRAXIS. The long double version of EPM resulted in Euclidean distances 6.019E-06 (*maxfun* = 1) and 2.154E-11 (*maxfun* = 0). The absolute values of the differences between the smallest Euclidean distances under PRAXIS and EPM were 2.593E-01 (*maxfun* = 1) and 1.190E-10 (*maxfun* = 0).

Summary: These results indicated that even with such a unimodal function slight improvements could be achieved using the EPM approach. This was especially true for setting '$maxfun = 1$' (e.g., see long double versions).

B. Modified wave function

The modified wave function is based on the sine and cosine functions altered by a bell-shaped function. It is a two-parameter function ($x, y \in \mathbb{R}$)

$$g(x, y) := \exp\left(-\frac{(x - \frac{5}{4}\pi)^2 + (y - \frac{1}{4}\pi)^2}{20} \right) \cdot \left(\sin(x+y) + \cos(x-y) \right),$$

which has a (unique) global minimum -2 at point ((5/4)π, (1/4)π), with various local minima. A Mathematica plot of this function is depicted in Figure 2.

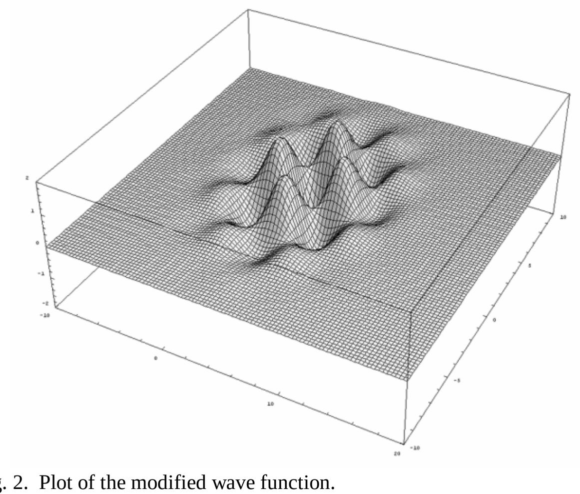

Fig. 2. Plot of the modified wave function.

TABLE I

MINIMIZATION RESULTS FOR THE ROSENBROCK FUNCTION

Double version, maxfun = 1

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	5.990257138849760E-02	1.018301802095190E+00	3.420357271720240E-04	4.098569120465070E-02
	1.860718519161620E+00	1.036672481830270E+00		
PRAXIS	5.990257138849760E-02	1.059722401501310E+00	3.566765242372130E-03	1.367427630163140E-01
	1.860718519161620E+00	1.123011454735940E+00		
PRAXIS	8.566400000000000E+02	8.536400000000000E+02	5.264567074311800E+13	3.241054773017570E+03
	3.125890000000000E+03	3.127890000000000E+03		

Double version, maxfun = 0

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	2.997958114835880E+00	1.000000001082150E+00	1.171066722436550E-18	2.419769930378260E-09
	-3.491414211106350E-01	1.000000002164310E+00		
PRAXIS	2.997958114835880E+00	1.000000001082150E+00	1.172603864548440E-18	2.423276700426840E-09
	-3.491414211106350E-01	1.000000002168230E+00		
PRAXIS	8.566400000000000E+02	5.874360983932630E+01	3.334337413888250E+03	3.450306297249240E+03
	3.125890000000000E+03	3.450823070877760E+03		

Long double version, maxfun = 1

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	2.529086281435810E+00	1.000002691874950E+00	7.246190765643310E-12	6.019221859832670E-06
	-7.754252305662780E-01	1.000005383757150E+00		
PRAXIS	2.529086281435810E+00	1.129664363507910E+00	2.827263823354180E-01	2.593196371856940E-01
	-7.754252305662780E-01	1.224574769433720E+00		
PRAXIS	-7.178368032444260E+01	-6.878368032444260E+01	2.234725392149370E+09	6.984439495385250E+01
	1.911608978852530E+00	3.911608978852530E+00		

Long double version, maxfun = 0

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	4.217765044704490E+00	1.000000000009630E+00	9.283854970219120E-23	2.154224020857930E-11
	1.595741204590470E+00	1.000000000019270E+00		
PRAXIS	4.217765044704490E+00	9.999999999699350E-01	3.899038824864630E-18	1.405531120156460E-10
	1.595741204590470E+00	1.000000000137300E+00		
PRAXIS	-3.342191534662190E+00	1.000000009322850E+00	3.711604876093000E-14	3.901879148096910E-08
	1.480348412059320E+00	1.000000037888660E+00		

For EPM, we used the following settings (cp. Section II): C = 0 (center point), d = 5 (primary interval), e = 105 (secondary intervals), w = 0.800 (weight), and c = 1.000E-14 (Round 3 difference criterion); twenty random start value vectors were generated in Round 1, twenty iterations were performed in Round 2, and twenty iterations were specified as the maximum number of iterative minimization runs in Round 3. The variable '*maxfun*' is a global variable of the PRAXIS implementation. It is an integer which specifies the maximum number of calls to the function to be minimized. PRAXIS returns after (at the latest) '*maxfun*' calls. The value 0 indicates no limit on the number of calls.

[1] For EPM, the start values were the random start values that resulted in the smallest minimum in Round 1. In particular, we ran PRAXIS with these start values (obtained from Round 1 of EPM), and with a second collection of start values.

[2] Euclidean distance of the vector of parameter estimates to the point (1, 1).

In Table II, we list the minimization results for this function.

Double versions: Minima ranged from -1.424E+00 to 0.000E+00 (*maxfun* = 1) and -2.000E+00 to -1.096E-20 (*maxfun* = 0) using the double version of the PRAXIS implementation with different start values. The double version of EPM resulted in minima -2.000E+00 (*maxfun* = 1) and -2.000E+00 (*maxfun* = 0). The absolute values of the differences between the smallest approximated minima under PRAXIS and EPM were 5.755E-01 (*maxfun* = 1) and 0.000E+00 (*maxfun* = 0).

The Euclidean distances of the vectors of parameter estimates to the point ((5/4)π, (1/4)π) at which the global minimum is attained ranged from 7.671E-01 to 4.492E+02 (*maxfun* = 0) and 1.221E-09 to 3.016E+01 (*maxfun* = 0) for the double version of PRAXIS. The double version of EPM resulted in Euclidean distances 8.454E-10 (*maxfun* = 1) and 1.982E-10 (*maxfun* = 0). The absolute values of the differences between the smallest Euclidean distances under PRAXIS and EPM were 7.671E-01 (*maxfun* = 1) and 1.023E-09 (*maxfun* = 0).

Long double versions: The long double version of PRAXIS resulted in minima which ranged from -1.725E+00 to -1.237E-13 (*maxfun* = 1) and -2.000E+00 to -2.275E-04 (*maxfun* = 0). The long double version of EPM gave the same minimum -2.000E+00 for both settings '*maxfun* = 1' and '*maxfun* = 0'. The absolute values of the differences between the smallest approximated minima under PRAXIS and EPM were 2.747E-01 (*maxfun* = 1) and 0.000E+00 (*maxfun* = 0).

The Euclidean distances of the vectors of parameter estimates to the point ((5/4)π, (1/4)π) at which the global minimum is attained ranged from 5.080E-01 to 2.460E+01 (*maxfun* = 1) and 6.190E-09 to 1.304E+01 (*maxfun* = 0) for

TABLE II

MINIMIZATION RESULTS FOR THE MODIFIED WAVE FUNCTION

Double version, maxfun = 1

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	4.398236202401620E+00	3.926990817831740E+00	-2.000000000000000E+00	8.453504466059670E-10
	3.341134781562980E+00	7.853981633595390E-01		
PRAXIS	4.398236202401620E+00	3.398236202401620E+00	-1.424466392025100E+00	7.670884116030240E-01
	3.341134781562980E+00	1.341134781562980E+00		
PRAXIS	4.265600000000000E+02	4.248528932188130E+02	0.000000000000000E+00	4.491865172424720E+02
	1.568900000000000E+02	1.575971067811860E+02		

Double version, maxfun = 0

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	4.302408341155940E+00	3.926990816793300E+00	-2.000000000000000E+00	1.981687160710880E-10
	3.202161325188850E+00	7.853981634381680E-01		
PRAXIS	4.302408341155940E+00	3.926990818103010E+00	-2.000000000000000E+00	1.220691149229250E-09
	3.202161325188850E+00	7.853981638925680E-01		
PRAXIS	4.265600000000000E+02	-2.623797823897020E+01	-1.095854667145390E-20	3.016496907234920E+01
	1.568900000000000E+02	7.863926045602940E-01		

Long double version, maxfun = 1

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	3.432054288785910E+00	3.926990816950400E+00	-2.000000000000000E+00	4.512712822560600E-11
	-1.159983751401300E+00	7.853981634235110E-01		
PRAXIS	3.432054288785910E+00	4.432054288785910E+00	-1.725271026026150E+00	5.080081158568010E-01
	-1.159983751401300E+00	8.400162485986980E-01		
PRAXIS	-2.239485859870910E+01	-2.067363633594930E+01	-1.236773346883290E-13	2.460084478485790E+01
	2.888876716114780E+00	8.888767161147840E-01		

Long double version, maxfun = 0

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]
EPM	2.572503715654020E+00	3.926990821201380E+00	-2.000000000000000E+00	6.189739009687640E-09
	-3.178412083810750E-01	7.853981679310880E-01		
PRAXIS	2.572503715654020E+00	3.926990821201380E+00	-2.000000000000000E+00	6.189739009687640E-09
	-3.178412083810750E-01	7.853981679310880E-01		
PRAXIS	6.803586005233220E+01	1.562968360357100E+01	-2.275105679323870E-04	1.304366401144220E+01
	4.840096754450370E+00	6.545966918844460E+00		

For EPM, we used the following settings (cp. Section II): $C = 0$ (center point), $d = 5$ (primary interval), $e = 105$(secondary intervals), $w = 0.900$ (weight), and $c = 1.000E\text{-}14$ (Round 3 difference criterion); twenty random start value vectors were generated in Round 1, twenty iterations were performed in Round 2, and twenty iterations were specified as the maximum number of iterative minimization runs in Round 3. The variable '*maxfun*' is a global variable of the PRAXIS implementation. It is an integer which specifies the maximum number of calls to the function to be minimized. PRAXIS returns after (at the latest) '*maxfun*' calls. The value 0 indicates no limit on the number of calls.

[1] For EPM, the start values were the random start values that resulted in the smallest minimum in Round 1. In particular, we ran PRAXIS with these start values (obtained from Round 1 of EPM), and with a second collection of start values.

[2] Euclidean distance of the vector of parameter estimates to the point $((5/4)\pi, (1/4)\pi)$.

the long double version of PRAXIS. The long double version of EPM resulted in Euclidean distances 4.513E-11 (*maxfun* = 1) and 6.190E-09 (*maxfun* = 0). The absolute values of the differences between the smallest Euclidean distances under PRAXIS and EPM were 5.080E-01 (*maxfun* = 1) and 0.000E+00 (*maxfun* = 0).

Summary: For setting '*maxfun* = 0', PRAXIS reported the true minimum in both of its (double and long double) versions; for '*maxfun* = 1', it resulted in minima relatively far away from the true minimum, in any version. The EPM procedure yielded the true minimum for the two settings in both versions. Finally, though PRAXIS and EPM each reported the true minimum in their double versions for setting '*maxfun* = 0', the Euclidean distance of the vector of parameter estimates to the point $((5/4)\pi, (1/4)\pi)$ was smaller for EPM than for PRAXIS. These results again indicated that slight improvements could be achieved using the EPM approach.

C. LCMRE function

The LCMRE (*Latent Class Model with Random Effects*) minus log (kernel of) likelihood function (briefly, LCMRE function) in Ünlü [4] is a more complex, empirical function based on binary response data. For this function, any information about extrema is missing. Such a kind of function is more difficult to minimize; however, this is closer to optimization problems in realistic situations.

In [4], the LCMRE function is applied for the estimation of response error (careless error and lucky guess) probabilities when subjects respond to dichotomous test items.

The LCMRE function is defined as

$$h(\theta;x) := -\sum_{R\in 2^Q}\left\{N(R)\cdot\ln\left(\sum_{u=0}^{1}\left[\tau_u\sum_{j=1}^{20}\left(w_j\prod_{l=1}^{m}\left[\Phi(a_{lu}+b_u t_j)^{s_l(R)}\cdot(1-\Phi(a_{lu}+b_u t_j))^{1-s_l(R)}\right]\right)\right]\right)\right\},$$

where

(1) $Q := \{I_l : l = 1, 2, ..., m\}$ is a set of $m \in \mathbb{N}_{\geq 1}$ dichotomously scored test items – a correct answer is scored 1, and an incorrect answer 0 –, and 2^Q its power-set;

(2) $R \in 2^Q$ stands for a response pattern (set of all items a subject solves), and $s_l(R) \in \{0,1\}$ $(1 \leq l \leq m)$ are the l^{th} entries of R's representation as m-list of 0's and 1's;

(3) the data, $x := (N(R))_{R \in 2^Q}$, are constituted by the observed absolute counts $N(R) \in \mathbb{N}_{\geq 0}$ of response patterns $R \in 2^Q$;

(4) $\theta := (a_{10}, a_{20}, ..., a_{m0}, a_{11}, a_{21}, ..., a_{m1}, b_0, b_1, \tau_1)$ is the parameter vector to be estimated, ranging over the parameter space $\Theta := \mathbb{R}^{2m+2} \times (0,1)$ (note, $\tau_0 + \tau_1 = 1$);

(5) $\Phi : \mathbb{R} \to [0,1]$ is the cumulative distribution function of a unit normal variate;

(6) $\{(t_j \in \mathbb{R}, w_j \in \mathbb{R}_{>0}) : j = 1, 2, ..., 20\}$ is the set of 20^{th} order Gauss-Hermite quadrature points.

Based on the traditional unrestricted 2-classes latent class model, we simulated a binary (of type $0/1$) 1500×7 data matrix representing the response patterns for 1500 fictitious subjects to *m = 7* test items. These data were used for the analyses to follow.

Double versions: Our comparison showed that minimizing the LCMRE function was not possible using the double versions of PRAXIS and EPM. Apparently, this was due to rounding errors that resulted in undefined operations (e.g., division by zero), and consequently, in undefined values for a minimum.

Long double versions: Using the long double version of PRAXIS, we obtained minima

5.297099870217230E+03 ($maxfun = 1$),

5.055607199977240E+03 ($maxfun = 0$).

The long double version of EPM resulted in minima

5.247813256004320E+03 ($maxfun = 1$),

5.055607181239010E+03 ($maxfun = 0$).

The absolute values of the differences between the smallest approximated minima under PRAXIS and EPM were 4.929E+01 ($maxfun = 1$) and 1.874E-05 ($maxfun = 0$).

The Euclidean distances between the vectors of parameter estimates under PRAXIS and EPM were

4.541479081653670E+01 ($maxfun = 1$),

2.469200259958340E+00 ($maxfun = 0$).

Summary: The long double conversions of PRAXIS and EPM were important for this complex, empirical function; minimization of the LCMRE function was simply not possible using their double versions. The results indicated that EPM could facilitate and improve the minimization of a complex, empirical function.

In all three applications, a comparison of processing times between PRAXIS and EPM's Round 2 and Round 3 showed no significant differences.

IV. Conclusion

The three rounds of EPM seem to enable and facilitate the finding of smaller minima and more stable parameter estimates, while processing times do not increase significantly. As well, the long double conversions appeared to be crucial for complex, difficult to minimize empirical functions. In summary, EPM did not yield worse results than PRAXIS in all minimization trials, and it seems to be a reasonable automation of numerical function minimization using the PRAXIS implementation. This may especially be true for complex, empirical minimization problems. The C source files are freely available from the authors.

Acknowledgment

This work was supported by the Austrian Science Fund (FWF) grant P17071-N04 to Dietrich Albert. The authors wish to thank Dietrich Albert for his valuable comments.

References

[1] K. R. Gegenfurtner, "PRAXIS: Brent's algorithm for function minimization," *Behavior Research Methods, Instruments, & Computers*, 24, pp. 560-564, 1992.

[2] R. P. Brent, *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[3] M. J. D. Powell, "An efficient method of finding the minimum of a function of several variables without calculating derivatives," *Computer Journal*, 7, pp. 155-162, 1964.

[4] A. Ünlü, "Estimation of careless error and lucky guess probabilities for dichotomous test items by means of latent class modeling with random effects," *Journal of Mathematical Psychology*, submitted for publication.