IMPLEMENTATION AND APPLICATIONS OF A THREE-ROUND USER STRATEGY FOR IMPROVED PRINCIPAL AXIS MINIMIZATION[1, 2]

Ali ÜNLÜ[3]

PhD, Assistant Professor, Department of Computer Oriented Statistics and Data Analysis, Institute of Mathematics, University of Augsburg, Germany

E-mail: ali.uenlue@math.uni-augsburg.de

Michael D. KICKMEIER-RUST[4]

MSc, Senior Project Coordinator,
Department of Psychology, University of Graz, Austria

E-mail: michael.kickmeier@uni-graz.at

Abstract: *This paper presents a three-round user strategy (EPM), extending the C implementation of Brent's PRAXIS algorithm by Gegenfurtner. In a first round, EPM applies a multistart procedure for global optimization, randomly generating and evaluating multiple sets of start values drawn from weighted primary and secondary intervals. Using the parameter estimates of the smallest first round minimum, in a second and third round, EPM performs iterative minimization runs and applies an additional break-off criterion to improve and stabilize the approximated minimum and parameter estimates. Moreover, EPM increases the precision of the original PRAXIS implementation by a conversion from the double to the long double data type. This conversion is not trivial and even seen to be essential for minimizing a complex empirical function from psychometrics. Important special cases of EPM are discussed and promising strategies for the handling of EPM are proposed. EPM's advantages over PRAXIS are illustrated using two different functions: a 'well-behaved' Rosenbrock function and an 'ill-behaved' psychometric likelihood function.*

Key words: Evaluation and improvement of optimization software; Extended principal axis minimization; Numerical optimization; PRAXIS; Psychometrics

Introduction

Remarks on the PRincipal AXIS (PRAXIS) minimization

Minimizing functions of several variables is a common and important problem in psychology and in natural sciences in general. A popular method is the *PRincipal AXIS* (PRAXIS) minimization by Brent (1973), an algorithm for numerical minimization of multivariable functions without the use of derivatives. PRAXIS is a modification of Powell's (1964) direction-set method; for a review of PRAXIS, see Gegenfurtner (1992). Gegenfurtner (1992) also provides an implementation of Brent's algorithm in the programming language C which is freely available on the Internet (see Section Availability).

The PRAXIS algorithm has been applied to a variety of problems in psychology; for instance, see Hartinger (1999) and Regenwetter, Falmagne, and Grofman (1999) in decision making, D'Zmura, Rinner, and Gegenfurtner (2000) and Heller (2001) in visual perception, or Doignon and Falmagne (1985, 1999), Fries (1997) and Ünlü (2006) in the psychometric theory of knowledge spaces. Further references on PRAXIS applications in psychological sciences are Erdfelder and Buchner (1998) in process-dissociation modeling, Heller (1997) in psychophysics, and McClelland and Chappell (1998) in memory research. PRAXIS type algorithms have been also applied in fields other than psychology; for instance, see Carcione, Mould, Pereyra, Powell, and Wojcik (2001) in computational acoustics, Hwang and Tien (1996) in physics, Leroy, Mozer, Payan, and Troccaz (2004) in medical image computing, Ren, Chen, Wu, and Yang (2002) and Ren, Wu, Yang, and Chen (2002) in nuclear medicine, Tubic, Zaccarin, Beaulieu, and Pouliot (2001) in medical physics, or Woelk (2000) in magnetic resonance.

Basic motivations for an Extended Principal axis Minimization (EPM)

In a realistic context, optimizing an objective function, in general, is more than to use a computer and a software application (e.g., PRAXIS), implicitly and unjustifiably assuming that (a) the actual optimization exercise utilizing the software is trivial and (b) the obtained results are accurate (cf. McCullough & Vinod, 1999). Quite the contrary, (a) optimization problems in practice often concern functions with many local extrema rather than being globally convex, concave, or otherwise 'good-natured.' Moreover, (b) numerical optimization algorithms and their software implementations differ in quality (e.g., accuracy of results), and one software application is not as good as any other, in general. This, however, is mostly hard to assess, especially for software users merely interested in application.

These points, in particular, apply to Brent's algorithm and its C implementation by Gegenfurtner. (a) PRAXIS generally converges to local minima; hence, minimization results strongly depend on the selection of suitable start values (cf. Table 1). Thus, in order to be also able to handle complex optimization problems, procedures for global optimization are required. Global optimization strategies for the PRAXIS algorithm have not been investigated, implemented and freely supplied so far, although there is a large body of strategies that could be considered. Methods are, for example, controlled random search (Price, 1983), evolutionary and genetic algorithms (Back & Schwefel, 1993), or multistart methods (Törn & Zilinskas, 1989). For a review of these approaches, see Pintér (1995). In this paper, we extend the original PRAXIS algorithm to *Extended Principal axis Minimization* (EPM) to offer a natural, flexible multistart procedure for global optimization. This provides a facilitation of usage resulting in improved and effective applications of PRAXIS (PRAXIS, currently, being merely available as a 'bare' C function).

Moreover, (b) the accuracy of the PRAXIS algorithm can be improved by three natural techniques: iterative minimization runs with reset program settings, an additional break-off criterion besides the PRAXIS algorithm's default criterion, and a conversion from the double to the long double data type. The EPM strategy proposed in this paper implements these techniques.

In the iterative part, EPM successively uses resulting parameter estimates as vectors of start values for new runs of the complete PRAXIS routine with reset internal program settings. The use of a second break-off criterion based on a minimal change in approximated minima, while PRAXIS applies a default criterion based on a minimal change in parameter estimates, supplements the iterative part to offer the possibility of improving and stabilizing the approximated minimum and parameter estimates. Finally, to gain more computational precision for complex optimization problems, EPM uses the long double data type instead of the double data type of the original PRAXIS implementation.[5]

A conversion from the double to the long double data type, however, is not trivial. It cannot be accomplished by simply altering the declaration/definition of variables in the original C source files (e.g., 'long double tol' instead of 'double tol'). A conversion requires the use of long double functions instead of double functions (e.g., 'long double sqrtl(long double)' instead of 'double sqrt(double)'), and it should also consider changing the data type of relevant constants (e.g., '0.0L' instead of '0.0'), throughout the entire source files. On the other hand, the conversion from the double to the long double data type is even seen to be essential for minimizing the empirical psychometric function in Subsection LCMRE function. Optimization of the *Latent Class Model with Random Effects* (LCMRE) likelihood function, which is based on real psychological test data, was not possible using the double (data type) 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 or parameter estimate. Use of the long double data type simply accommodated this problem.

In the end, the different components of EPM can be combined with each other to give important special cases of the EPM strategy in practice. Additionally enhanced by a multiplicity of intra-component strategies for the handling of individual components, this offers a great flexibility of usage in the actual minimization exercise utilizing the EPM extension (cf. Fig. 3).

Additional notes

Similar to optimization heuristics such as *simulated annealing*, *genetic algorithms*, or *ant colonies* (for details and further references, see, e.g., Winker, 2001, and Winker & Gilli, 2004), EPM allows 'uphill moves', that is, larger minima in consecutive iterations than in previous ones. Throughout its three rounds, however, EPM stores the smallest minimum found and the corresponding parameter estimates globally, and outputs them as the final solution (cf. Fig. 2).

EPM is implemented in C programming language (ANSI C 99). The function is held very general, so it can be easily used with other algorithms available in C/C++ without extensive modifications. The source code for EPM is freely available from the authors.

It shall be also noted that the long double (data type) versions of PRAXIS and EPM do not run on Windows 32 systems (95, 98, NT, Me, 2000, XP) properly. On these systems, the long double data type is directly mapped to the double data type. To overcome this limitation on Windows systems, extensive workarounds are necessary.

Architecture of EPM

The EPM strategy consists of three rounds which are described next. In all rounds, the user can additionally specify the PRAXIS settings.[6] An overview of the EPM strategy is schematically shown in Fig. 2. In Section Examples, EPM and the original PRAXIS routine are implemented using both the double and the long double data type.

Round 1

In Round 1, EPM applies a variant of a multistart procedure to cover global optimization. Such procedures generally produce a set of random start value vectors and evaluate an objective function at these vectors. Then a number of start value vectors with the lowest function values are selected from this initial set of vectors and local search methods are applied. EPM uses a variant of such a multistart procedure. A number of start value vectors are randomly generated based on a specific 'three-interval-uniform-sampling' design. An objective function is evaluated at these vectors using the PRAXIS algorithm. The parameter estimates of the best candidate vector of start values resulting in the smallest minimum are then locally investigated in subsequent rounds of EPM.

More precisely, in Round 1, a number $N_1 \in \mathbb{N} := \{1, 2, \ldots\}$ of start value vectors are randomly generated based on user specified, weighted primary and secondary intervals (see Fig. 1). The primary interval is determined by a center point C and a number *d* > 0 as [C – *d*, C + *d*]. Two secondary intervals surrounding the primary interval are specified by a number e > *d* as [C – e, C – *d*) and (C + *d*, C + e]. The probability for sampling a value from the primary interval is specified by a weight $0 \leq w \leq 1$, and consequently, the probability for sampling a value from any of the two secondary intervals is set to (1 – *w*) / 2 each. For all intervals, start values are randomly drawn using uniform distributions. That is, the density function for the primary interval is $f_p(t) := \begin{cases} 1/2d & \text{for } C-d \leq t \leq C+d \\ 0 & \text{else} \end{cases}$, and for the secondary intervals they are given by $f_{s1}(t) := \begin{cases} 1/(e-d) & \text{for } C-e \leq t < C-d \\ 0 & \text{else} \end{cases}$ and $f_{s2}(t) := \begin{cases} 1/(e-d) & \text{for } C+d < t \leq C+e \\ 0 & \text{else} \end{cases}$, respectively (*t*, a real number). The constants C, *d*, e, and *w* can be chosen from the set $\mathbb{R}$ of real numbers, within the limitations of the respective data type used in the implementation.[5] The objective function is evaluated at each of the N_1 randomly generated start value vectors by performing a minimization run with PRAXIS. The parameter estimates of the best candidate vector of start values resulting in the smallest minimum are then locally investigated in subsequent Rounds 2 and 3 of the EPM strategy.

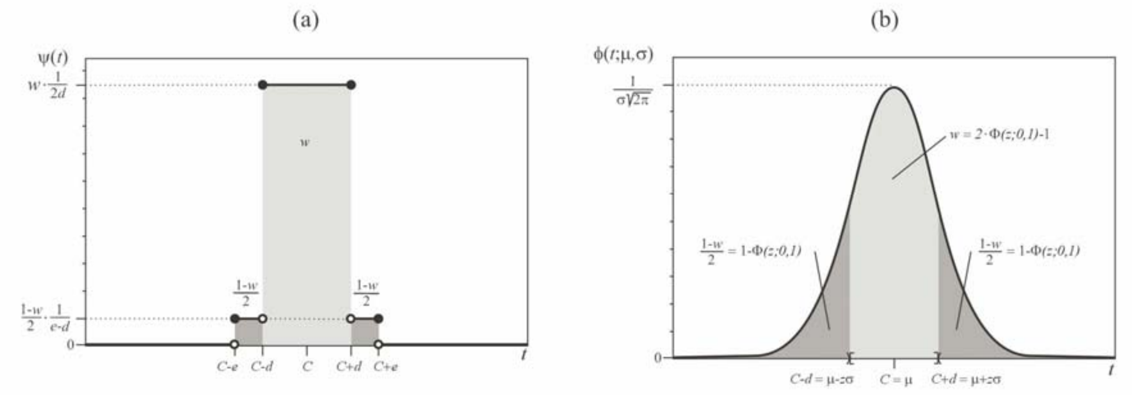

Figure 1. Random sampling of start values based on (a) the three-interval-uniform-sampling design and (b) the normal distribution

(a) *Three-interval-uniform-sampling design*: For all intervals, start values are randomly drawn using uniform distributions. Given f_p the density function for the primary interval, and f_{s1} and f_{s2} the density functions for the secondary intervals, let $\psi := w \cdot f_p + \frac{1-w}{2} \cdot \left(f_{s1} + f_{s2} \right)$.

(b) *Normal distribution*: Start values are randomly drawn using the Gaussian density $\phi(.;\mu,\sigma)$ with mean μ and standard deviation σ. To sample a value from the primary interval with a given probability (weight) *w* (strictly between zero and one), we have to solve the equation $w = 2 \cdot \Phi(z;0,1) - 1$ for z, where $\Phi(.;\mu,\sigma)$ is the Gaussian cumulative distribution function to $\phi(.;\mu,\sigma)$. Consequently, the probability for sampling a value from any of the two secondary intervals is (1 – w) / 2 each. Given this weight distribution over the primary and secondary intervals, every required spread *d* of the primary interval can be achieved by the choice of σ; simply solve the equation $d = z\sigma$ for σ (for given *d* and z).

Some remarks are in order with respect to this sampling design.

1. In a clear and straightforward manner, this design represents a natural modeling approach to capturing subjective confidences PRAXIS users may have in promising parameter regions for global minima. This modeling approach is gradual, in the sense that stronger subjective confidence regions can be covered by primary intervals while weaker ones are covered by secondary intervals, differentiated by the choices of weights. A strategy for a concrete specification of this sampling design, of course, strongly depends on the properties of an objective function and the prior knowledge about the latter. For instance, if no prior information about a function is available, a promising strategy may be the specification of a large number of start value vectors, an extensive primary interval, large secondary intervals, and weak PRAXIS settings[2] (to reduce computational efforts). If there is, however, prior knowledge of the region for a global minimum, it may be promising to focus on that region with a narrow primary interval (*d* small), negligible secondary intervals (e close to zero), and a weight close to unity.

2. If we set the weight to unity, as an important special case in practice, we obtain uniform sampling from the primary interval only; secondary intervals are no longer considered. The width of this single sampling (primary) interval can be further gradually sharpened up by limiting the constant *d* to zero. This allows a great flexibility in narrowing down promising parameter regions. An overview of the special cases of the EPM strategy is schematically shown in Fig. 3.

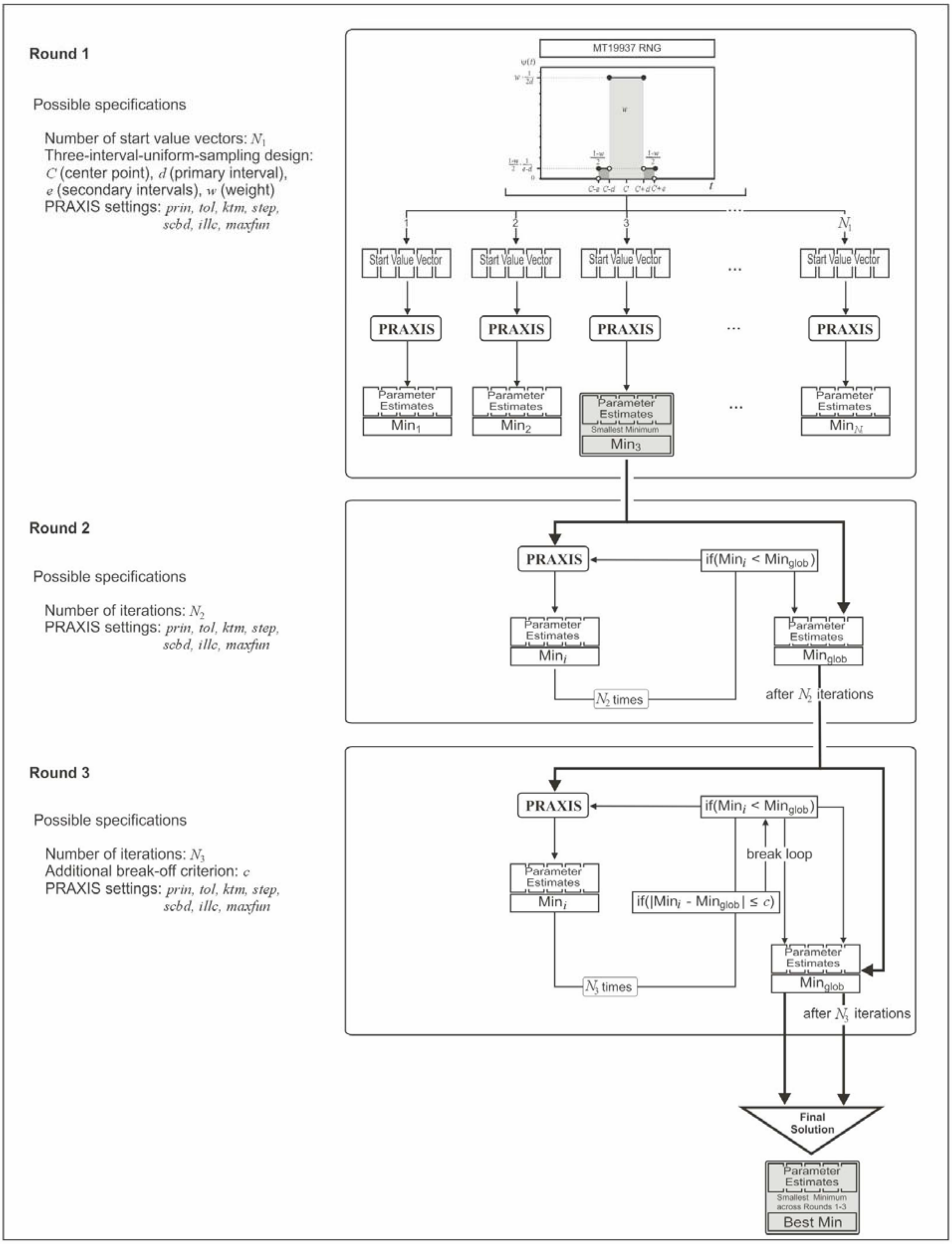

Figure 2. Overview of the EPM strategy. The EPM strategy can be implemented using both the double and the long double data type. 'MT19937 RNG' stands for the Mersenne Twister 19937 (uniform pseudo) random number generator

3. There are, of course, alternative sampling designs for the random generation of start value vectors (see Section Discussion). One alternative, for instance, is based on the normal distribution (with fixed weight distribution over the primary and secondary intervals). In this alternative, the user has to specify only two constants (mean as the center point and standard deviation as the spread) instead of having to specify the constants C, *d*, *e*, and *w* of the three-interval-uniform-sampling approach. Such alternative designs, however, generally come with a loss of modeling of subjective confidences for promising parameter regions. In general, this modeling is not that clear, straightforward, and natural anymore under such alternatives.

4. For the implementation of probabilities, EPM utilizes the *Mersenne Twister* (MT) 19937. MT19937 is a uniform pseudorandom number generator which was developed by Matsumoto and Nishimura (1998). It provides fast generation of high quality random numbers (e.g., period $2^{19937}-1$), and rectifies many of the flaws found in older generators (e.g., Wichmann-Hill random number generator; Wichmann & Hill, 1982, 1984). MT19937 is freely available on the Internet (see Section Availability).

Round 2

The best vector of parameter estimates in Round 1 is the most promising candidate for further minimization analyses in its neighborhood. Based on these parameter estimates, in Round 2, EPM performs iterative loops successively reapplying resulting parameter estimates as start values for new runs of the PRAXIS routine. This offers the possibility of improving minimization results. The best Round 2 results are then subject to stabilization analyses in Round 3.

More precisely, in Round 2, EPM performs a number $N_2 \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ of iterative minimization runs. Starting with the best parameter estimates of Round 1, the resulting parameter estimates of each iteration are passed to the next iteration as start values for a new run of the PRAXIS routine. Though PRAXIS is internally based upon an iterative procedure, running the complete PRAXIS routine afresh with reset program settings according to this iterative paradigm generally improves the minimization results. The obtained minimum and parameter estimates can be steadily improved, allowing refined, closer approximations to the true values. The best Round 1 minimum and the corresponding vector of parameter estimates are globally stored in variables m_{glob} and v_{glob}, respectively. In each of the N_2 iterations, the current minimum m_{cur} and the corresponding vector of parameter estimates v_{cur} are globally stored in m_{glob} and v_{glob}, respectively, if $m_{cur} < m_{glob}$. This assures that the smallest minimum and the corresponding vector of parameter estimates found across all previous runs are stored, even if a slightly growing minimum is obtained for an iteration (which may occur since PRAXIS applies a minimal change in parameter estimates as the default break-off criterion; as mentioned in Subsection Additional notes, EPM allows 'uphill moves'). Round 2 finally closes with the smallest minimum found so far, and the corresponding parameter estimates. These results are then subject to stabilization analyses based on an additional break-off criterion in Round 3.

Some remarks are in order with respect to this iterative paradigm.

1. A strategy for Round 2 depends on the complexity of an objective function and the computational effort required minimizing it. The more complicated an objective function is to minimize, the more processing time is required if a larger number of iterations and

stronger PRAXIS settings are specified in Round 2. A promising strategy could be, for instance, the selection of a larger number of iterations and weak to medium PRAXIS settings.

2. Running PRAXIS afresh with reset settings according to this iterative paradigm generally improves the obtained results. In Section Examples smaller minima could be obtained for about 20 to 30 iterative runs. This was especially true for weaker PRAXIS settings.

3. The EPM strategy offers a great flexibility of usage. As a special case of EPM, the number of iterations in Round 2 can be set to zero, so the user is able to merely apply the PRAXIS routine with the multistart (global optimization) procedure of Round 1 and the stabilization stage (additional break-off criterion) in Round 3. Additionally setting the number of iterations in Round 3 to zero yields another important special case of the EPM strategy, PRAXIS enhanced for global optimization by the multistart procedure of Round 1 only. For an overview of the special cases of the EPM strategy, see Fig. 3.

Round 3

In Round 3, the iterative loops of minimization runs are supplemented by the application of an additional break-off criterion. The purpose of Round 3 is to stabilize the minimization results. The best Round 3 minimum and the corresponding parameter estimates, that is, the best minimization results across all three rounds of EPM, are output as the final solution of the EPM strategy.

More precisely, as in Round 2, in Round 3 EPM performs a maximum number $N_3 \in \mathbb{N}_0$ of iterative minimization runs. Starting with the globally stored best parameter estimates obtained in Round 2, the resulting parameter estimates of each iteration are passed to the next iteration as start values for a new minimization run. In each iteration, the current minimum (m_{cur}) and the corresponding vector of parameter estimates are globally stored if $m_{cur} < m_{glob}$. In contrast to Round 2, however, this time the iterative part is combined with the application of an additional break-off criterion. While PRAXIS internally applies a minimal change in estimated parameter vectors as the default break-off criterion (see Gegenfurtner, 1992), EPM introduces a (in general small) number $c \geq 0$ as an additional break-off criterion based on a minimal change in the approximated minimum m_{cur} of a current run and the globally stored smallest minimum m_{glob} found across all previous runs so far. If $\left| m_{cur} - m_{glob} \right| \leq c,$ EPM stops the iterative loops. As long as changes larger than the criterion *c* occur, upwards or downwards, EPM continues the iterative loops, until the maximum number of iterations N_3 is reached. That way the results can be stabilized in general, in the sense that the iterations stop when the obtained minimum does not vary anymore, except for minimal changes quantified by a small *c*. Round 3, and in particular, EPM, finally close with the smallest minimum and the corresponding parameter estimates found across all three rounds. These best results of the minimization exercise utilizing EPM are globally stored (m_{glob} and v_{glob}, respectively) and output as the final solution of the EPM strategy.

Some remarks are in order with respect to this stabilization stage.

1. A strategy for Round 3 depends on the computational effort required minimizing an objective function. The more complicated an objective function is to minimize, the more processing time is required if a larger number of iterations, a stricter break-off criterion *c*, and stronger PRAXIS settings are specified in Round 3. A promising strategy could be, for

instance, the selection of a smaller number of iterations, a strict break-off criterion, and strong PRAXIS settings.

2. Why do we continue the iterative procedure in Round 3, combined with the use of an additional break-off criterion not already applied in Round 2? This iterative procedure in general improves the obtained results (see Section Examples). Applying the additional break-off criterion already in Round 2 would have the disadvantage of generally stopping the iterations at an earlier stage of steady improvement; hence, yielding only suboptimal results in general. Therefore, the iterative loops are continued and the break-off criterion is first applied in Round 3, after the 'burn-in' iterations in Round 2. The break-off criterion then captures whether gained improvements stabilize in best results (except for minimal variations quantified by the criterion).

3. As a special case of EPM, the number of iterations in Round 3 can be set to zero, so the user is able to apply the PRAXIS routine with the multistart procedure of Round 1 and the iterative procedure (without the additional break-off criterion) in Round 2 only (see Fig. 3).

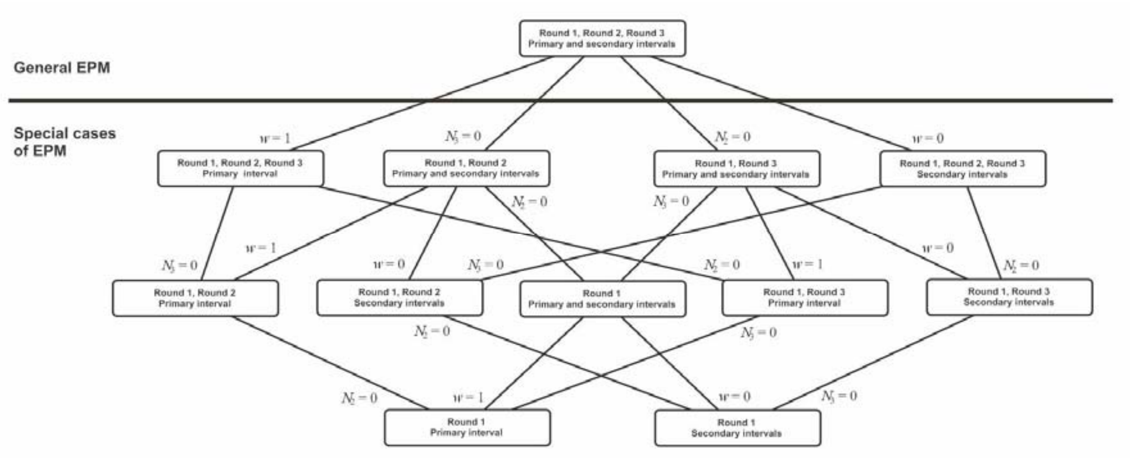

Figure 3. Diagram of the special cases of the EPM strategy

Transitivity is not explicitly depicted in the diagram; for instance, 'Round 1 / Primary interval' is a special case of 'Round 1, Round 2 / Primary and secondary intervals' (imposed restrictions: w = 1 and $N_2 = 0$). As an example of a special case of the general EPM strategy (vertex 'Round 1, Round 2, Round 3 / Primary and secondary intervals'), the number of iterations in Round 2 can be set to zero (edge $N_2 = 0$), so the user is able to merely apply the PRAXIS routine with the multistart procedure of Round 1 and the stabilization stage in Round 3 (vertex 'Round 1, Round 3 / Primary and secondary intervals'). Additionally setting the number of iterations in Round 3 to zero (edge $N_3 = 0$) yields another important special case, PRAXIS enhanced for global optimization by the multistart procedure of Round 1 only (vertex 'Round 1 / Primary and secondary intervals').

Examples

EPM improves the original PRAXIS implementation by four extensions: the automatic generation and evaluation of random start value vectors (multistart procedure for global optimization), the iterative loops to approach the true minimum and parameter values, an additional break-off criterion to stabilize minimization results, and a conversion from the double to the long double data type.

To illustrate EPM's advantages over the original PRAXIS implementation, we ran basic minimization trials using two different functions: a Rosenbrock function and a complex empirical function from psychometrics. We contrasted results obtained based on different start value vectors (for different specifications of the three-interval-uniform-sampling design), PRAXIS settings (*maxfun* = 0 and *maxfun* = 1), and double as well as long double data type versions of both the EPM and PRAXIS implementations.

Apart from the *maxfun* setting, in each case we used the following PRAXIS settings: *tol* = 1.000E-20, *ktm* = 1, *step* = 1.000, and *scbd* = 1.000. In each case the EPM strategy generated $N_1 = 20$ start value vectors (for different specifications of the sampling design), performed $N_2 = 20$ Round 2 and a maximum number $N_3 = 20$ Round 3 iterations, and applied an additional break-off criterion *c* = 1.000E-14. All computations were performed on a laptop computer with a Pentium II 366 MHz processor and 256 MB RAM running a LINUX system (SUSE LINUX 9.0).

Rosenbrock function

The Rosenbrock function in Fig. 4 is often considered as a test problem for optimization algorithms. It is a two-variable unimodal function ($x, y \in \mathbb{R}$)

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

which has a unique global minimum 0 at the point (1, 1).

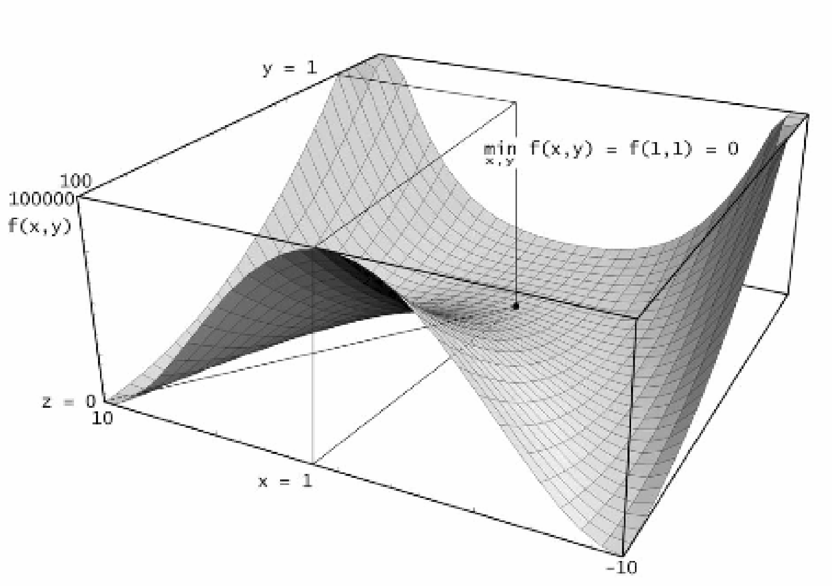

Figure 4. Plot of the Rosenbrock function

At any location other than (1, 1) this function has no local extremum. We will, however, make the following important observation (concerning the robustness of the PRAXIS routine against 'bad' choices of start values): Using vectors of start values distant to (1, 1) will cause PRAXIS to misleadingly converge and stop the minimization process at, in each case, different locations (not only far from the true location) the Rosenbrock function has no local minima at all.

For EPM's sampling design in Round 1, in each case we used the following constants: *C* = 0.000 (center point), *d* = 5.000 (primary interval), *e* = 105.000 (secondary intervals), and *w* = 0.800 (weight).

Results for the double versions: First, we consider the results obtained for the double versions of EPM and PRAXIS (see Table 1).

For the *maxfun* = 1 setting, EPM yielded a minimum of 3.420E-04. The Euclidean distance of the vector of parameter estimates to the point (1, 1) at which the global minimum is attained was 4.099E-02. For the original PRAXIS implementation we applied two different vectors of start values. First, we ran PRAXIS with the start values that resulted in the smallest minimum in EPM's Round 1; in this condition, PRAXIS yielded a minimum of 3.567E-03. The Euclidean distance of the vector of parameter estimates to the point (1, 1) was 1.367E-01. Second, we used a start value vector (8.566E+02, 3.126E+03) more distant to the location of the global minimum; in this condition, PRAXIS yielded a minimum of 5.265E+13. The Euclidean distance of the vector of parameter estimates to the coordinates of the global minimum was 3.241E+03.

For the *maxfun* = 0 setting, EPM yielded a minimum of 1.171E-18. The corresponding Euclidean distance was 2.420E-09. Again, for PRAXIS we applied two different start value vectors. Running PRAXIS with the start values that resulted in the smallest minimum in EPM's Round 1 yielded a minimum of 1.173E-18, with the corresponding Euclidean distance 2.423E-09. A start value vector (8.566E+02, 3.126E+03) distant to the point (1, 1) resulted in a minimum of 3.334E+03. The Euclidean distance of the vector of parameter estimates to (1, 1) was 3.450E+03.

Results for the long double versions: In a second step, we consider the results obtained for the long double versions of EPM and PRAXIS (see Table 1).

For *maxfun* = 1, EPM yielded a minimum of 7.246E-12. The corresponding Euclidean distance was 6.019E-06. Using the start values that resulted in the smallest minimum in EPM's Round 1, PRAXIS yielded a minimum of 2.827E-01. The Euclidean distance of the vector of parameter estimates to the point (1, 1) was 2.593E-01. A start value vector (-7.178E+01, 1.912E+00) more distant to the location of the global minimum resulted in a minimum of 2.235E+09, with the corresponding Euclidean distance 6.984E+01.

For *maxfun* = 0, EPM yielded a minimum of 9.284E-23. The corresponding Euclidean distance was 2.154E-11. Using the start values that resulted in the smallest minimum in EPM's Round 1, PRAXIS yielded a minimum of 3.899E-18. The corresponding Euclidean distance was 1.406E-10. Running PRAXIS with a second set of start values (-3.342E+00, 1.480E+00) yielded a minimum of 3.712E-14. The Euclidean distance of the vector of parameter estimates to the point (1, 1) was 3.902E-08.

Table 1. Minimization results for the Rosenbrock function

Method	Start Values[1]	Parameter Estimates	Minimum	Euclidean Distance[2]	t[3]
Double version using maxfun = 1					
EPM	5.990257138849760E-02	1.018301802095190E+00	3.420357271720240E-04	4.098569120465070E-02	<1
	1.860718519161620E+00	1.036672481830270E+00			
PRAXIS	5.990257138849760E-02	1.059722401501310E+00	3.566765242372130E-03	1.367427630163140E-01	<1
	1.860718519161620E+00	1.123011454735940E+00			
PRAXIS	8.566400000000000E+02	8.536400000000000E+02	5.264567074311800E+13	3.241054773017570E+03	<1
	3.125890000000000E+03	3.127890000000000E+03			
Double version using maxfun = 0					
EPM	2.997958114835880E+00	1.000000001082150E+00	1.171066722436550E-18	2.419769930378260E-09	<1
	-3.491414211106350E-01	1.000000002164310E+00			
PRAXIS	2.997958114835880E+00	1.000000001082150E+00	1.172603864548440E-18	2.423276700426840E-09	<1
	-3.491414211106350E-01	1.000000002168230E+00			
PRAXIS	8.566400000000000E+02	5.874360983932630E+01	3.334337413888250E+03	3.450306297249240E+03	<1
	3.125890000000000E+03	3.450823070877760E+03			
Long double version using maxfun = 1					
EPM	2.529086281435810E+00	1.000002691874950E+00	7.246190765643310E-12	6.019221859832670E-06	<1
	-7.754252305662780E-01	1.000005383757150E+00			
PRAXIS	2.529086281435810E+00	1.129664363507910E+00	2.827263823354180E-01	2.593196371856940E-01	<1
	-7.754252305662780E-01	1.224574769433720E+00			
PRAXIS	-7.178368032444260E+01	-6.878368032444260E+01	2.234725392149370E+09	6.984439495385250E+01	<1
	1.911608978852530E+00	3.911608978852530E+00			
Long double version using maxfun = 0					
EPM	4.217765044704490E+00	1.000000000009630E+00	9.283854970219120E-23	2.154224020857930E-11	<1
	1.595741204590470E+00	1.000000000019270E+00			
PRAXIS	4.217765044704490E+00	9.999999999699350E-01	3.899038824864630E-18	1.405531120156460E-10	<1
	1.595741204590470E+00	1.000000000137300E+00			
PRAXIS	-3.342191534662190E+00	1.000000009322850E+00	3.711604876093000E-14	3.901879148096910E-08	<1
	1.480348412059320E+00	1.000000037888660E+00			

Note: Apart from the *maxfun* setting, in each case we used the following PRAXIS settings: *tol* = 1.000E-20, *ktm* = 1, *step* = 1.000, and *scbd* = 1.000. In each case the EPM strategy generated $N_1 = 20$ Round 1 start value vectors, performed $N_2 = 20$ Round 2 and a maximum number $N_3 = 20$ Round 3 iterations, and applied an additional break-off criterion c = 1.000E-14. All computations were performed on a laptop computer with a Pentium II 366 MHz processor and 256 MB RAM running a LINUX system (SUSE LINUX 9.0). In each case we used the following constants for EPM's three-interval-uniform-sampling design in Round 1: C = 0.000 (center point), *d* = 5.000 (primary interval), e = 105.000 (secondary intervals), and *w* = 0.800 (weight).

[1]For EPM, the start values are 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) at which the unique global minimum 0 is attained.

[3]Processing time in seconds. Note that for EPM this is the processing time required across all three rounds.

As summarized in Table 1, EPM achieved (partially substantial) smaller minima and Euclidean distances than PRAXIS in all conditions. The same results were even found on the level of individual parameter estimates (i.e., not only for aggregate Euclidean distances). In any condition, EPM yielded individual parameter estimates closer to 1 than the original PRAXIS routine (cf. also the remark in Subsection LCMRE function). All this was especially true for the *maxfun* = 1 setting, under both the double and the long double versions. Finally, processing times were below 1 second across all conditions.

In particular, we made the following important observation concerning the robustness of the PRAXIS routine against 'bad' choices of start values. The results in Table 1 pointed out that PRAXIS was not robust against choices of start values more distant to the location of the global minimum. Using vectors of start values distant to (1, 1) caused PRAXIS to misleadingly converge and stop the minimization process at, in each case, different locations (not only far from the true location) the Rosenbrock function has no local minima at all. This turned out to be the case for both the double and the long double version of PRAXIS, under any of the settings *maxfun* = 1 and *maxfun* = 0. This may underline the importance of the multistart procedure implemented in EPM's Round 1 (cf. Table 1).

LCMRE function

The *Latent Class Model with Random Effects* (LCMRE) function is a negative log (kernel of) likelihood function which is derived based on a probit regression latent class modeling with random effects (for details, see Ünlü, 2006). It is a complex empirical function which has been originally proposed by Qu, Tan, and Kutner (1996) for the estimation of the accuracy (sensitivity and specificity) of a diagnostic test for screening individuals in biometrics (see also Hadgu & Qu, 1998; Hui & Zhou, 1998; Qu & Hadgu, 1998). Recently, Ünlü (2006) has adapted and applied this approach for the estimation of response error (careless error and lucky guess) probabilities when examinees respond to dichotomous test items in the psychometric theory of knowledge spaces.

The LCMRE function depends on the observed binary response data, and any information about extrema of this function is lacking. Such kind of 'ill-behaved' function is more difficult to minimize. However, this is close to optimization problems in practice.

Briefly, the LCMRE function is defined as

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

where

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

(2) $R \in 2^Q$ denotes the set of test items solved by a subject (response pattern), and $s_l(R) \in \{0,1\}$ $(1 \le l \le m)$ 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 represented by the observed absolute counts $N(R) \in \mathbb{N}_0$ of the 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 which ranges over the parameter space $\Theta := \mathbb{R}^{2m+2} \times (0,1)$, and we have $\tau_0 + \tau_1 = 1$;

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

(6) $\left\{(t_j \in \mathbb{R}, w_j \in \mathbb{R}_{>0}) : j = 1, 2, \ldots, 20\right\}$ is a set of (known) constants obtained from the 20th order Gauss-Hermite quadrature.

Based on the classical unrestricted 2-classes latent class model (the psychological model assumed to underlie the responses of a subject; see Ünlü, 2006) we simulated a binary (of type 0/1) 1500×7 data matrix representing the response patterns of 1500 fictitious subjects to *m* = 7 test items. These data and the C source code for simulating these data are freely available from the first author. This data set was the basis for the subsequent analyses.

Results for the double versions: First, we consider the results obtained for the double versions of EPM and PRAXIS (see Table 2).

Computations using the double versions were not possible for the LCMRE function; we tried a great many start value vectors from a great many parameter regions, and a great many specifications of the EPM strategy. Apparently, due to rounding errors by limitations of the double data type, we received undefined operations, and consequently, undefined minima or parameter estimates. Thus, in this example, the conversion from the double to the long double data type was essential for making minimization of that psychometric function possible at all. Use of the long double data type accommodated this problem easily.

Results for the long double versions: In a second step, we consider the results obtained for the long double versions of EPM and PRAXIS (see Table 2).

Computations using the long double versions could be performed. For EPM's sampling design in Round 1, in each case we used the following constants: C = 0.000 (center point), *d* = 0.200 (primary interval), *e* = 0.500 (secondary intervals), and *w* = 0.900 (weight). Because we do not have any information about extrema of the LCMRE function, we computed the Euclidean distances between the vectors of $2 \cdot 7 + 3 = 17$ parameter estimates under EPM and PRAXIS.

For the *maxfun* = 1 setting, EPM yielded a minimum of 5.248E+03 and PRAXIS (using EPM's Round 1 best vector of start values) resulted in a minimum of 5.297E+03. The Euclidean distance of the EPM vector of parameter estimates to the PRAXIS vector of parameter estimates was 4.541E+01.

For the *maxfun* = 0 setting, EPM yielded a minimum of 5.05560718**1239010**E+**03** and PRAXIS (using EPM's Round 1 best vector of start values) resulted in a minimum of 5.0556071**99977240**E+**03**. The corresponding Euclidean distance was 2.469E+00.

Table 2. Minimization results for the LCMRE function

Method	Minimum	Euclidean Distance[1]	t[2]
Double version using maxfun = 1			
EPM	1.#QNAN	_	<1
PRAXIS	1.#QNAN		<1
Double version using maxfun = 0			
EPM	1.#QNAN	_	<1
PRAXIS	1.#QNAN		<1
Long double version using maxfun = 1			
EPM	5.247813256004320E+03	4.541479081653670E+01	624
PRAXIS	5.297099870217230E+03		17
Long double version using maxfun = 0			
EPM	5.055607181239010E+03	2.469200259958340E+00	845
PRAXIS	5.055607199977240E+03		18

Note: Apart from the *maxfun* setting, in each case we used the following PRAXIS settings: *tol* = 1.000E-20, *ktm* = 1, *step* = 1.000, and *scbd* = 1.000. In each case the EPM strategy generated $N_1 = 20$ Round 1 start value vectors, performed $N_2 = 20$ Round 2 and a maximum number $N_3 = 20$ Round 3 iterations, and applied an additional break-off criterion *c* = 1.000E-14. All computations were performed on a laptop computer with a Pentium II 366 MHz processor and 256 MB RAM running a LINUX system (SUSE LINUX 9.0). Computations using the double versions were not possible (indicated by '1.#QNAN' and '–'). Due to rounding errors by limitations of the double data type, we received undefined operations. Computations using the long double versions, however, could be performed. For the long double versions, in each case we used the following constants for EPM's three-interval-uniform-sampling design in Round 1: C = 0.000 (center point), *d* = 0.200 (primary interval), *e* = 0.500 (secondary intervals), and *w* = 0.900 (weight). In particular, we ran PRAXIS with the random start values that resulted in the smallest minimum in Round 1 of EPM.

[1]Euclidean distance of the EPM vector of parameter estimates to the PRAXIS vector of parameter estimates.

[2]Processing time in seconds. Note that for EPM this is the processing time required across all three rounds.

As summarized in Table 2, the conversion from the double to the long double data type turned out to be important for the psychometric LCMRE function; minimizing this function using the double versions of EPM and PRAXIS was not possible.

Computations, however, could be performed using the long double versions. EPM achieved (partially substantial) smaller minima than PRAXIS in both conditions *maxfun* = 1 and *maxfun* = 0. In particular, we found parameter estimates under EPM and PRAXIS clearly deviating from each other, even in the *maxfun* = 0 condition. Finally, processing times using EPM were 624 and 845 seconds in the conditions *maxfun* = 1 and *maxfun* = 0, respectively, contrary to PRAXIS with 17 (*maxfun* = 1) and 18 (*maxfun* = 0) seconds.

An important remark is in order with respect to deviations in the parameter estimates. Though the approximated minima in the condition *maxfun* = 0 differed only minimally (compared to the larger deviations in the condition *maxfun* = 1), the parameter estimates yielded a clearer difference. This, however, may be a crucial factor in practical applications. In general, a user is primarily interested in the optimizing parameter estimates. The LCMRE function, for instance, is used for maximum likelihood estimation of response error probabilities for dichotomous test items in psychometrics. These probabilities, however, are functions of the parameter estimates (see Ünlü, 2006). In particular, deviations in the parameter estimates may occur and be empirically important (in the current example, resulting in better maximum likelihood estimates for the response error rates of the underlying classical unrestricted 2-classes latent class model), even if the approximated minima differ only slightly. Hence, from an empirical point of view, (even little) improvements in approximated minima may be desirable, and thus longer processing times may be acceptable.

Discussion

Summary

We have introduced, implemented and applied the three-round EPM strategy to improve the original PRAXIS implementation by four extensions (cf. Fig. 2): a multistart procedure to cover global optimization, iterative loops to approach the true minimum and parameter values, an additional break-off criterion to stabilize minimization results, and a conversion from the double to the long double data type to increase computational precision for complex optimization problems. We have also seen that this strategy offers a number of important special cases in practice (cf. Fig. 3), and thus provides the user with a great flexibility in the actual minimization exercise utilizing the EPM extension.

EPM's advantages over the original PRAXIS implementation have been illustrated using two different functions: a 'well-behaved' Rosenbrock function (see Fig. 4) for which the global minimum and its corresponding coordinates are known, and an 'ill-behaved' complex empirical function from psychometrics for which any information about extrema is lacking. For both functions, across all conditions, EPM improved (partially substantial) the minimization results obtained using merely the original PRAXIS implementation (see Tables 1 and 2). Processing times, however, increased for the LCMRE function using EPM. Nevertheless, in practical applications, (even little) improvements in approximated minima may be worthwhile (e.g., yielding better maximum likelihood estimates for empirically interpreted functions of the parameter estimates), and thus longer processing times may be acceptable.

We have observed that a not necessarily trivial (see Subsection Basic motivations for an Extended Principal axis Minimization (EPM)) conversion from the double to the long double data type can significantly improve computational precision. As demonstrated with the LCMRE function, for some complex empirical minimization problems the long double data type may be essential for performing principal axis minimization based on the original PRAXIS implementation. For such problems, computational values may exceed the limitations of the double data type, resulting in undefined operations (e.g., division by zero), and consequently, yielding undefined results (see Table 2). Moreover, for the Rosenbrock function we have observed that the original PRAXIS implementation was not robust against 'bad' choices of start values. Using start value vectors more distant to the location of the global minimum caused PRAXIS to misleadingly converge and stop the minimization process at, in each case, different locations not only far from the true location, but at which the Rosenbrock function has no local minima at all. The EPM strategy easily accommodated these two observations by implementing double as well as long double versions (cf. Table 2) and a multistart procedure for global optimization (cf. Table 1), respectively.

Further extensions and modifications

As mentioned in Subsection Round 1, there are, of course, other sampling designs for the random generation of start value vectors than the three-interval-uniform-sampling approach applied in this paper. One alternative could be based on the family of two-parameter normal distributions (Gaussian densities)

$$\phi:\mathbb{R}\rightarrow[0,+\infty),\ t\mapsto\phi(t;\mu,\sigma):=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right)\quad(\mu\in\mathbb{R},\sigma\in\mathbb{R}_{>0})$$

(cf. Fig. 1). In this alternative, the user could specify two constants: the mean as the center point (i.e., $C:=\mu$) and the standard deviation as the spread (i.e., $d:=\sigma$). Then start values could be sampled based on this normal distribution. Compared to the three-interval-uniform-sampling design, the primary interval (interpreted as the stronger subjective confidence region) could be defined as $[\mu-\sigma,\mu+\sigma]$, while the secondary intervals (interpreted as the weaker subjective confidence regions) could be represented by $(-\infty,\mu-\sigma)$ and $(\mu+\sigma,+\infty)$. Under these conditions, the probability for sampling a start value from the primary interval is $2\cdot\Phi(1;0,1)-1\approx 0.682$, and consequently, the probability for sampling a start value from any of the two secondary intervals is $1-\Phi(1;0,1)\approx 0.159$ each ($\Phi(.;\mu,\sigma)$ Gaussian cumulative distribution function corresponding to $\phi(.;\mu,\sigma)$). These probabilities, however, are the same for any specification of the primary and secondary intervals, that is, for any choice of the constants (center point) $C:=\mu$ and (spread) $d:=\sigma$.

In order to imitate the weight of the three-interval-uniform-sampling design allowing a flexible distribution of unit mass over the primary and secondary intervals, one could, for instance, more generally define the primary interval as $[\mu-z\sigma,\mu+z\sigma]$ and the secondary intervals by $(-\infty,\mu-z\sigma)$ and $(\mu+z\sigma,+\infty)$; here z is any positive real number. The center point again is $C:=\mu$, the spread however is given by $d:=z\sigma$. Then the probability for sampling a start value from the primary interval is $2\cdot\Phi(z;0,1)-1$, while the probability for sampling a start value from any of the two secondary intervals is $1-\Phi(z;0,1)$ each. Since $\lim_{t\rightarrow+\infty}\Phi(t;0,1)=1$ and $\lim_{t\rightarrow 0+}\Phi(t;0,1)=1/2$, we can control, by the choice of z, the probabilities for sampling start values from the primary and secondary intervals. Given any such weight distribution over the intervals, every required width (i.e., spread *d*) of the primary interval can be achieved by the choice of σ (cf. Fig. 1).

Moreover, what has been said for the normal distribution could also be applied (maybe with minor modifications) with any family of two-parameter probability distributions in which one parameter represents a location parameter (to place the distribution at a location on a parameter axis) and the other a shape parameter (to control the shape/mass of the distribution around that location). Finally, these alternatives, and the three-interval-uniform-sampling approach, can be further generalized to respectively include a normal distribution (a two-parameter probability distribution) and a three-interval-uniform-sampling design for each model parameter separately. In this case, constants of the sampling designs are indexed by the model parameters.

We have also outlined an analysis of the different factors varied in this paper. Future work may address systematic variations of other factors as well; for instance, in the case of empirical functions, variations of sample size (especially for small sample sizes) and model complexity, or other PRAXIS settings (than *maxfun*).

Conclusion

The EPM strategy represents a wrapper around the function minimization routine PRAXIS. It effectively improves the original PRAXIS implementation by techniques that are applicable with other routines as well. Future work could include implementing this strategy for other routines in use; in particular, it would be interesting to see whether it provides similar advantages with, for instance, the functions for minimization in Mathematica and Matlab.

Availability

The C (ANSI C 99) source files, including EPM, PRAXIS, and MT19937, for double as well as long double versions, are freely available from the authors. Electronic mail may be sent to ali.uenlue@math.uni-augsburg.de or michael.kickmeier@uni-graz.at. The original PRAXIS source by Karl Gegenfurtner can be found at http://archives.math.utk.edu/software /msdos/numerical.analysis/praxis/.html (retrieved July 19, 2005).

The original source of the MT19937 pseudorandom number generator, which is a component of the EPM implementation, can be found at http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html (retrieved July 19, 2005). The C source file for simulating data using the classical unrestricted 2-classes latent class model and the data set simulated for the analyses in this paper can also be freely obtained from the first author by electronic mail.

References

1. Back, T. and Schwefel, H.-P. **An overview of evolutionary algorithms for parameter optimization,** Evolutionary Computation, 1,1993, pp.1-23
2. Brent, R.P. **Algorithms for minimization without derivatives,** Englewood Cliffs, NJ, Prentice-Hall, 1973
3. Carcione, L., Mould, J., Pereyra, V., Powell, D. and Wojcik, G. **Nonlinear inversion of piezoelectrical transducer impedance data,** Journal of Computational Acoustics, 9, 2001, pp. 899-910
4. Doignon, J.-P. and Falmagne, J.-Cl. **Spaces for the assessment of knowledge,** International Journal of Man-Machine Studies, 23, 1985, pp. 175-196
5. Doignon, J.-P. and Falmagne, J.-Cl. **Knowledge spaces,** Berlin, Springer, 1999
6. D'Zmura, M., Rinner, O. and Gegenfurtner, K.R. **The colors seen behind transparent filters,** Perception, 29, 2000, pp. 911-926
7. Erdfelder, E. and Buchner, A. **Process-dissociation measurement models: Threshold theory or detection theory?,** Journal of Experimental Psychology: General, 127, 1998, pp. 83-96
8. Fries, S. **Empirical validation of a Markovian learning model for knowledge structures,** Journal of Mathematical Psychology, 41, 1997, pp. 65-70
9. Gegenfurtner, K.R. **PRAXIS: Brent's algorithm for function minimization,** Behavior Research Methods, Instruments, and Computers, 24, 1992, pp. 560-564
10. Hadgu, A. and Qu, Y. **A biomedical application of latent class models with random effects,** Applied Statistics, 47, 1998, pp. 603-616
11. Hartinger, A. **Do generalized expected utility theories capture persisting properties of individual decision makers?**, Acta Psychologica, 102, 1999, pp. 21-42
12. Heller, J. **On the psychophysics of binocular space perception,** Journal of Mathematical Psychology, 41, 1997, pp. 29-43
13. Heller, J. **Mittenbildung bei achromatischen Farben: Das klassische Experiment von Plateau [Bisection with achromatic colors: Plateau's classical experiment],** Zeitschrift für Experimentelle Psychologie, 48, 2001, pp. 259-271

14. Hui, S.L. and Zhou, X.H. **Evaluation of diagnostic tests without gold standards,** Statistical Methods in Medical Research, 7, 1998, pp. 354-370
15. Hwang, J.S. and Tien, C. **XRAYSCAN: An indexing program considering dense spurious peaks in an optimization method,** Chinese Journal of Physics, 34, 1996, pp. 47-57
16. Leroy, A., Mozer, P., Payan, Y. and Troccaz, J. **Rigid registration of freehand 3D ultrasound and CT-scan kidney images,** in MICCAI 2004, PT 1, "Proceedings Lecture Notes in Computer Science", Vol. 3216, 2004, pp. 837-844
17. Matsumoto, M. and Nishimura, T. **Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator,** ACM Transactions on Modeling and Computer Simulation, 8, 1998, pp. 3-30
18. McClelland, J.L. and Chappell, M. **Familiarity breeds differentiation: A subjective-likelihood approach to the effects of experience in recognition memory,** Psychological Review, 105, 1998, pp. 724-760
19. McCullough, B.D. and Vinod, H.D. **The numerical reliability of econometric software,** Journal of Economic Literature, 37, 1999, pp. 633-665
20. Pintér, J.D. **Global optimization in action,** Dordrecht, Kluwer, 1995
21. Powell, M.J.D. **An efficient method for finding the minimum of a function in several variables without calculating derivatives,** Computer Journal, 7, 1964, pp. 155-162
22. Price, W.L. **Global optimization by controlled random search,** Journal of Optimization Theory and Applications, 40, 1983, pp. 333-348
23. Qu, Y. and Hadgu, A. **A model for evaluating sensitivity and specificity for correlated diagnostic tests in efficacy studies with an imperfect reference test,** Journal of the American Statistical Association, 93, 1998, pp. 920-928
24. Qu, Y., Tan, M. and Kutner, M.H. **Random effects models in latent class analysis for evaluating accuracy of diagnostic tests,** Biometrics, 52, 1996, pp. 797-810
25. Regenwetter, M., Falmagne, J.-Cl. and Grofman, B. **A stochastic model of preference change and its application to 1992 presidential election panel data,** Psychological Review, 106, 1999, pp. 362-384
26. Ren, H.P., Chen, S.Z., Wu, W.K. and Yang, H. **PET and MR three-dimensional image registration based on second-order mutual information,** World Journal of Nuclear Medicine, 1(suppl. 2), 293, 2002
27. Ren, H.P., Wu, W.K., Yang, H. and Chen, S.Z. **High accuracy registration for three-dimensional brain images of PET and MRI,** Chinese Journal of Nuclear Medicine, 22, 2002, pp. 168-170
28. Törn, A. and Zilinskas, A. **Global optimization,** Berlin, Springer, 1989
29. Tubic, D., Zaccarin, A., Beaulieu, L. and Pouliot, J. **Automated seed detection and three-dimensional reconstruction. II. Reconstruction of permanent prostate implants using simulated annealing,** Medical Physics, 28, 2001, pp. 2272-2279
30. Ünlü, A. **Estimation of careless error and lucky guess probabilities for dichotomous test items: A psychometric application of a biometric latent class model with random effects,** Journal of Mathematical Psychology, 50, 2006, pp. 309-328
31. Wichmann, B.A. and Hill, I.D. **Algorithm AS 183: An efficient and portable pseudo-random number generator,** Applied Statistics, 31, 1982, pp. 188-190
32. Wichmann, B.A. and Hill, I.D. **Correction: Algorithm AS 183: An efficient and portable pseudo-random number generator,** Applied Statistics, 33, 1984, p.123,
33. Winker, P. **Optimization heuristics in econometrics,** Chichester, Wiley, 2001
34. Winker, P. and Gilli, M. **Applications of optimization heuristics to estimation and modelling problems,** Computational Statistics & Data Analysis, 47, 2004, pp. 211-223
35. Woelk, K. **Torus factor - the relationship between radiofrequency field and radial position in toroid cavity probes,** Journal of Magnetic Resonance, 146, 2000, pp. 157-164

---

[1] This work was supported by the Austrian Science Fund (FWF) project grant P17071-N04 to Dietrich Albert.

[2] Correspondence to: Ali Ünlü, Institute of Mathematics, University of Augsburg, D-86135 Augsburg, Germany Phone: +49 821 598-2236, Fax: +49 821 598-2200, Email: ali.uenlue@math.uni-augsburg.de

[3] Dr. Ali Ünlü is an assistant professor in the Department of Computer Oriented Statistics and Data Analysis of the Institute of Mathematics at the University of Augsburg, Germany. In the period 1993-2000, he studied mathematics and physics at the University of Technology Darmstadt, Germany. He received his diplom in mathematics in 2000

from the University of Technology Darmstadt. In the period 2000-2006, he studied quantitative psychology (primarily specializing in probabilistic knowledge space theory and item response theory) as a postgraduate and postdoctoral research assistant in the Cognitive Science Section of the Institute of Psychology at the University of Graz, Austria. In the period 2002-2004, he completed a doctoral study in applied mathematics at the Graz University of Technology, Austria, receiving the doctor of technical sciences with an interdisciplinary dissertation on statistical model selection in the psychometric theory of knowledge spaces.
His research focuses on the mathematical and statistical foundations in the quantitative behavioral sciences. His research interests are in the areas of knowledge space theory, nonparametric item response theory, latent class analysis, model selection, numerical optimization, data analysis, information visualization, and software development.
In October 2007, he has been appointed associate editor of the European Journal of Pure and Applied Mathematics.
Representative publications:
Ünlü, A. **Nonparametric item response theory axioms and properties under nonlinearity and their exemplification with knowledge space theory,** Journal of Mathematical Psychology, 51, 2007, pp. 383–400
Ünlü, A. **Estimation of careless error and lucky guess probabilities for dichotomous test items: A psychometric application of a biometric latent class model with random effects,** Journal of Mathematical Psychology, 50, 2006, pp. 309–328
Ünlü, A., & Albert, D. **The Correlational Agreement Coefficient CA - a mathematical analysis of a descriptive goodness-of-fit measure,** Mathematical Social Sciences, 48, 2004, pp. 281–314
Ünlü, A. (in press). **A note on monotone likelihood ratio of the total score variable in unidimensional item response theory,** British Journal of Mathematical and Statistical Psychology
Ünlü, A., & Sargin, A. (in press). **Interactive visualization of assessment data: The software package Mondrian,** Applied Psychological Measurement

[4] Michael D. Kickmeier-Rust is a psychologist and software developer. He is a senior project coordinator in the Cognitive Science Section of the Department of Psychology at the University of Graz, Austria. In the period 1998-2005, he studied psychology at the University of Graz and received his master degree in cognitive psychology and industrial psychology.
His research and development activities are concentrating primarily on intelligent, adaptive educational systems. In particular, he is working on the evolution of psycho-pedagogical frameworks and models of adaptivity on the macro and micro levels. This work includes individualized navigation within learning environments, individualized presentation of educational material, as well as non-invasive assessment and interventions. Since 2001, he is with the Cognitive Science Section at the Department of Psychology, University of Graz and working in international, cross-discipline research and development projects. Since 2006, his very focus is on the psychological aspects of learning with immersive digital computer games and the integration of intelligent educational technology in game-based learning. Since 2008, he is senior project coordinator of the FP7 ICT project 80Days that is dealing with game-based learning. Related fields of interest are human-computer or man-machine interaction and specific fields of aviation psychology, dealing with education and training.
Since 2004 he is vice-chair of the Austria Aviation Psychology Association.
Representative publications:
Ebner, M., Kickmeier-Rust, M., & Holzinger, A. **Utilizing Wiki-Systems in higher education classes: a chance for universal access?**, Universal Access in the Information Society, 2008
Kickmeier-Rust, M.D., Peirce, N., Conlan, O., Schwarz, D., Verpoorten, D., Albert, D. **Immersive Digital Games: The Interfaces for Next-Generation E-Learning?** In C. Stephanidis (Ed.), "Universal Access in Human-Computer Interaction. Applications and Services" (pp. 647-656). Lecture Notes in Computer Science, 4556/2007. Berlin: Springer, 2007
Kickmeier-Rust, M.D., Albert, D., Hockemeyer, C., & Augustin, T. **Not breaking the narrative: Individualized Competence Assessment in Educational Games,** In "Proceedings of the European Conference on Games-based Learning (ECGBL)", October 25-26, Paisley, Scotland, 2007
Kickmeier-Rust, M.D., Ebner, M., & Holzinger, A. **Wikis: Do they need usability engineering?,** In M. Pohl, A. Holzinger, R. Motschnig, & C. Swertz (Eds.), "M3 – Interdisciplinary aspects on digital media & education" (pp. 137-144). Vienna: Österreichische Computer Gesellschaft, 2006
Kickmeier-Rust, M.D., & Vormayr, E.M. (Eds.). **Lecture Notes of ISAP'5,** International Summer School on Aviation Psychology (Vol. 2), 2005

[5] The double data type has a range of $\pm$ 1.798E+308 and a precision of 15 decimal places. The long double data type has a range of $\pm$ 1.1E+4932 and a precision of 19 decimal places.

[6] PRAXIS settings: *prin* controls the printed output from the routine; *tol* is the tolerance used for the default break-off criterion; *ktm* specifies the number of times the default criterion must be fulfilled to stop the minimization process; *step* is a step-size variable; *scbd* is a scaling variable; *illc* specifies whether the problem is difficult to minimize (ill conditioned) – PRAXIS automatically sets *illc* to true if it finds the problem to be ill conditioned; *maxfun* specifies the maximum number of internal calls to the objective function – a value 0 indicates no limit on the number of function calls. For a more detailed explanation of the PRAXIS settings, see Gegenfurtner (1992).