Mathematical Aspects of Internal Magnitude Calibration

D. Hömberg
Astronomisches Institut
Westfälische Wilhelms-Universität
Münster, F.R. Germany

Abstract

The internal calibration method, introduced by Bunclark and Irwin, is presented as a non-linear optimization process. Two different approaches to solve the problem are presented, the conjugate gradient method, a deterministic optimization algorithm, and simulated annealing, belonging to the class of Monte Carlo methods. Tests with model data show that the conjugate gradient method is superior to simulated annealing in computational effort, if the latter is defined by the number of equivalent functional evaluations. Both methods lead to the same average residual of the objective function. Considering the simplicity of the algorithm, simulated annealing suggests itself for solving optimization problems of high complexity with small theoretical effort.

1 Introduction

The most attractive feature of *internal calibration* is that it uses the objects on the photographic plate itself as gauging objects. It overcomes the problems of varying plate sensitivity and the lack of standards on the plate area under consideration.

The method was first developed by Bunclark and Irwin (1983). Fundamental to it is the concept, that the light of all stars on a well-defined area of the photographic plate is scattered by the same point-spread function. This means, the intensity profiles of the stars all have the same shape in logarithmic measure, while additive constants take care of the difference in stellar magnitudes.

In continuation of an earlier publication (Hömberg 1988) this paper describes internal calibration in terms of a nonlinear optimization process. Two different minimizing algorithms are introduced and their results are compared using model data.

2 Data reduction

To extract the data needed for the internal calibration procedure, picture frames of the direct plate are used (Horstmann 1988). For all frames containing stars, the centre of density is computed and by discrete integration over concentric areas and subsequent normalization one-dimensional profiles are obtained, in the following referred to as *density profiles.*

3 Internal calibration as a nonlinear optimization process

Let k be the characteristic curve transforming density to logarithmic intensity, *i.e.* $\log I = k(d)$. As derived in Hömberg (1988),

$$k'(D_1)D_1'(r) - k'(D_2)D_2'(r) = 0 \tag{1}$$

is the defining criterion for an internal characteristic curve. This means, every function, solving Eqn. 1 for arbitrary density profiles D_1 and D_2 will be called an *internal characteristic curve.*

This definition implies a corollary:

Every linear transformation $\tilde{k}(d) = ak(d) + c$ $(a \neq 0)$ of an internal characteristic curve $k(d)$ again is an internal characteristic curve.

To define the optimization problem, the analytical representation of the characteristic curve by Honeycutt and Chaldu (1970) is used which is Eqn. 2:

$$\log I = a_1 d + a_2 \ln(\exp(bd^{c_1}) - 1) + a_3 \exp(bd^{c_2}) + a_4 \; . \tag{2}$$

Assuming $a_1 \neq 0$, Eqn. 2 may be written this way:

$$\log I = a_1 \left(d + \frac{a_2}{a_1} \ln(\exp(bd^{c_1}) - 1) + \frac{a_3}{a_1} \exp(bd^{c_2}) \right) + a_4 \; . \tag{2'}$$

Eqn. 2′ makes it quite clear, that any choice of parameters a_1 and a_4 will represent an internal characteristic curve, provided that the remaining parameters have been estimated correctly.

Thus, in order to get an unique parameter representation, one chooses a_1 and a_4 as constants and defines Eqn. 3 to be the general representation of the internal characteristic curve:

$$k(x, d) = 0.1d + x_1 \ln(\exp(x_3 d^{x_4}) - 1) + x_2 \exp(x_3 d^{x_5}) \; , \tag{3}$$

with the parameter vector $x \in \mathbb{R}^5$ being the parameter vector.

Let the density profiles be marked $D_i(r), 1 \leq i \leq m$, with m equal to the total number of density profiles used for the procedure. As they are only known at certain discrete values r_j of the radius, one defines

$$d_{i,j} = D_i(r_j) \quad , \qquad 1 \leq i \leq m, 1 \leq j \leq n_i$$

with

$$n_i = max\{j | \, D_i(r_j) \neq 0\}$$

defined to be the length of the density profile D_i. The first derivative of the density profile D_i is approximated by the symmetric difference quotient

$$\Delta_{i,j} = \frac{1}{2}(d_{i,j+1} - d_{i,j-1}) \quad , \qquad 2 \leq j \leq n_i - 1 \; .$$

Now solving the internal calibration problem means searching for a vector $x \in \mathbb{R}^5$, so that Eqn. 1 is fulfilled for all pairs of density profiles D_i and D_j $(i \neq j)$. Thus one takes an arbitrary profile, names it D_1 and defines Eqn. 4 to be the objective function

$$f(x) = \frac{1}{m} \sum_{i=2}^{m} \sum_{j=2}^{N_i - 1} \left(k'(x, d_{1,j}) \Delta_{1,j} - k'(x, d_{i,j}) \Delta_{i,j} \right)^2 , \tag{4}$$

with $N_i = min(n_i, n_1)$ being the minimal length of the density profiles D_1 and D_i.

In Eqn. 4 the differences between D_1 and all other profiles are summed up according to Eqn. 1. Hence finding a parameter vector x for which the objective function $f(x)$ takes its minimal value, hopefully close to zero, becomes equivalent to solving the internal calibration problem.

4 Remarks about the objective function

Before discussing algorithms for solving this optimization problem, some remarks have to be made about the objective function.

i) The choice of a reference profile D_1 is not difficult. On the one hand its length should not be too short in order to use as much information about the data as possible. On the other hand it should not be overexposed, because this would falsify the results.

ii) In order to obtain useful characteristic curves, all parameters have to be larger or equal to zero. Therefore the domain of the objective function has to be restricted to the set

$$\mathcal{S} = \{ x \in \mathbb{R}^5 | x_i \geq 0, 1 \leq i \leq 5 \} .$$

Instead of solving the constrained optimization problem

$$\text{minimize } f(x), x \in \mathcal{S} ,$$

another approach is used here. A new objective function is defined to be the sum of the old one and a penalty function

$$P(x) = \sum_{i=1}^{5} (\min(0, x_i))^2 .$$

Whenever one component of x becomes less than zero, the function $F(x)$ is penalized by an increase of its function value. Thus one has to solve the unconstrained optimization problem

$$\text{minimize } F(x) = f(x) + \mu P(x) \quad , \qquad x \in \mathbb{R}^5 . \tag{5}$$

The greater μ, the better the constrained problem will be approximated by the unconstrained one.

iii) The minimum of the objective function is not defined sharply. Perhaps it may be better described as a broadenend valley with slowly increasing gradient (*Fig. 1*).

In the next two paragraphs optimization algorithms for solving the internal calibration problem (*i.e.* for minimizing the objective function as defined in Eqn. 5) will be presented.

5 Conjugate gradient methods

The simplest algorithm for minimizing nonlinear functions is the method of steepest descent. It is based on the fact that the gradient of a function always points at the direction of the steepest growth of a function, which implies that the negative gradient points at the direction of steepest descent of a function.

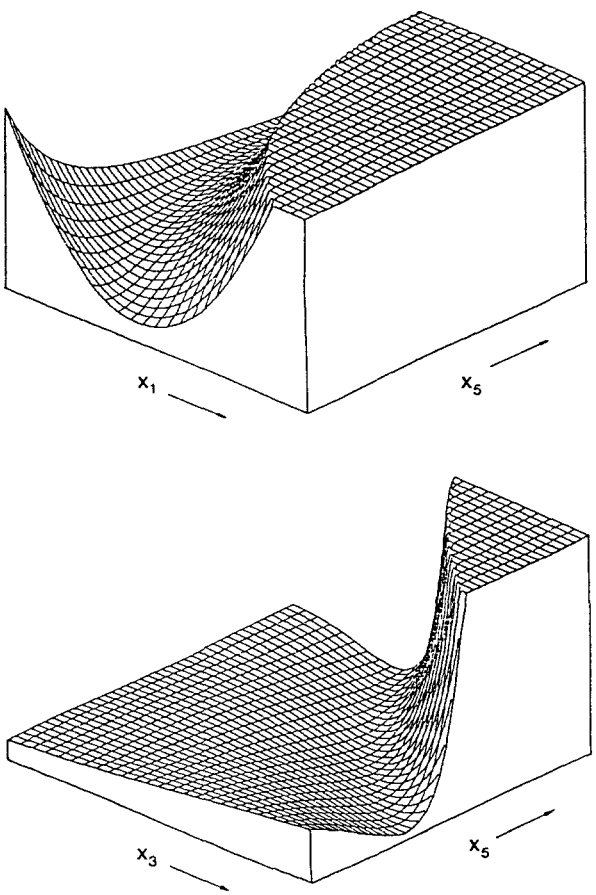

Fig. 1. Two-dimensional sections of the graph of the objective function in the neighbourhood of the minimum.

This means starting at a point x_k one searches along the direction of the negative gradient (grad $F)(x_k)$ to a minimum point of the line which is taken to be x_{k+1}. Thereby the five-dimensional minimization problem is reduced to a one-dimensional one which can easily be solved by standard line search techniques like quadratic and cubic curve fitting (*e.g.* Luenberger 1972).

The method of conjugate gradients differs from steepest descent only in the choice of the minimizing direction which is a linear combination of the old minimizing direction and the gradient. More precisely, the algorithm works as follows:

Conjugate Gradient Algorithm

Step I. Given x_0 compute $g_0 := (\text{grad } F)(x_0)$
set $d_0 := g_0$.

Step II. $x_{k+1} = x_k - \alpha_k d_k$, where α_k minimizes $F(x_k - \alpha d_k)$
$g_{k+1} = (\text{grad } F)(x_{k+1})$
$d_{k+1} = g_{k+1} + \beta_k dk$.

If $k+1$ is a multiple of 5, then

$$\beta_k = 0 ,$$

otherwise

$$\beta_k = \frac{\langle g_{k+1}, g_{k+1}\rangle}{\langle g_k, g_k\rangle} \qquad (FR)$$

or

$$\beta_k = \frac{\langle g_{k+1} - g_k, g_{k+1}\rangle}{\langle g_k, g_k\rangle} \qquad (PR)$$

or

$$\beta_k = \frac{\langle g_{k+1}, H_{k+1} d_k\rangle}{\langle d_k, H_{k+1} d_k\rangle} , \qquad (DAN)$$

with $H_k := (\text{Hess } F)(x_k)$ defined to be the Hessian of the objective function.

Hence every five steps the conjugate gradient method is restarted with a pure steepest descent step. Depending on the choice of β_k the algorithm is named Fletcher-Reeves (FR), Polak-Ribière (PR) or Daniel (DAN) method.

This comparatively simple algorithm is highly attractive due to its good convergence in the case of quadratic functions. Since every function in the neighbourhood of a minimum can be approximated by a quadratic function one expects that these properties can also be transferred to nonquadratic functions, at least in the neighbourhood of the minimum.

Unfortunately, the conjugate gradient method does not perform satisfactorily when applied to the internal calibration problem because of the ill-defined minimum of the objective function (Sect. 4). Furthermore this method is comparatively expensive in computing time. For each iteration, the gradient of the objective function, which is given through a very complex formula, has to be evaluated and the line search algorithm has to be called. Altogether, one iteration of the conjugate gradient method in connection with an efficient line search technique costs as much computing time as 21 evaluations of the objective function.

6 Simulated Annealing

In contrast to the deterministic optimization algorithm discussed in the last section, simulated annealing belongs to the class of Monte Carlo methods. Based on the so-called Metropolis Algorithm in statistical mechanics, it was developed independently by Kirkpatrick *et al.* (1983) and Cèrny (1985). The algorithm is very simple and is described in the following.

Simulated Annealing Algorithm

Step I. Choose starting point $x_0 \in \mathbb{R}^5$,
starting temperature $T_0 \in \mathbb{R}_+$,
cooling rate $q \in]0,1[$,
number of iterations
per fixed temperature $N \in \mathbb{N}$;
initialize $k = 0$
and $T_k = T_0$

Step II. Choose $\Delta x \in D$ 'at random', $D \subset \mathbb{R}^5$, being a compact cube centered at the origin
$\Delta F = F(x_k + \Delta x) - F(x_k)$
if $\Delta F \leq 0$ then
$x_{k+1} = x_k + \Delta x$
else
$P = \exp(-\frac{\Delta F}{T_k})$
choose $z \in [0,1]$ 'at random'
if $z \leq P$ then
$x_{k+1} = x_k + \Delta x$
else
$x_{k+1} = x_k$
end if
end if
$k = k + 1$

Step III. If k is a multiple of N then
$T_{k+1} = qT_k$
else
$T_{k+1} = T_k$
end if
go to step II.

After the initialization in step I has been carried out, for each iteration of step II a small Δx is chosen at random and the function value at the new point $x_k + \Delta x$ is computed. If this function value is smaller than the old one, $x_k + \Delta x$ is accepted to be x_{k+1}. Otherwise the worse value $x_k + \Delta x$ is accepted with the Boltzmann

probability P. This is realized by choosing a random number z between zero and one. If z is less or equal to P, then $x_k + \Delta x$ is accepted to be x_{k+1}, otherwise not.

After step II has been carried out N times, the temperature is lowered in step III. Then the algorithm starts again with step II. In other words, the probability to accept an increased function value at x_{k+1} decreases exponentially until the system freezes when T approaches zero.

The name Simulated Annealing is derived from the analogy to the cooling process of a physical system. In that case free energy assumes the role of the function to be minimized. If the system is cooled too fast, it freezes before reaching its ground state.

For a detailed introduction concerning mathematical theory and applications of simulated annealing the author refers to the textbook of van Laarhoven and Aarts (1987). An astronomical application is given by Jeffrey and Rosner (1986).

7 Discussion of results

For an objective comparison of both methods model data were used. A set of logarithmic Gaussian profiles (*i.e.* parabolas) were transformed back to two-dimensional density profiles. For a better representation of reality, Poisson noise was added to the profiles. Using the routines described in Sect. 2, two sets of one-dimensional density profiles were thus obtained (*Fig. 2*).

Table 1. Results for the conjugate-gradient-methods

a) model data without noise

algorithm	PR	FR	DAN
min. res.	0.02	0.01	0.04
aver. res.	0.05	0.05	0.08
# lt	11.5	12.6	5
# f	242	1264	195

b) model data with Poisson noise

algorithm	PR	FR	DAN
min. res.	0.37	0.37	0.38
aver. res.	0.46	0.49	0.51
# lt	8.4	7.6	6
# f	176	160	234

lt : average number of iterations

f : average number of function evaluations

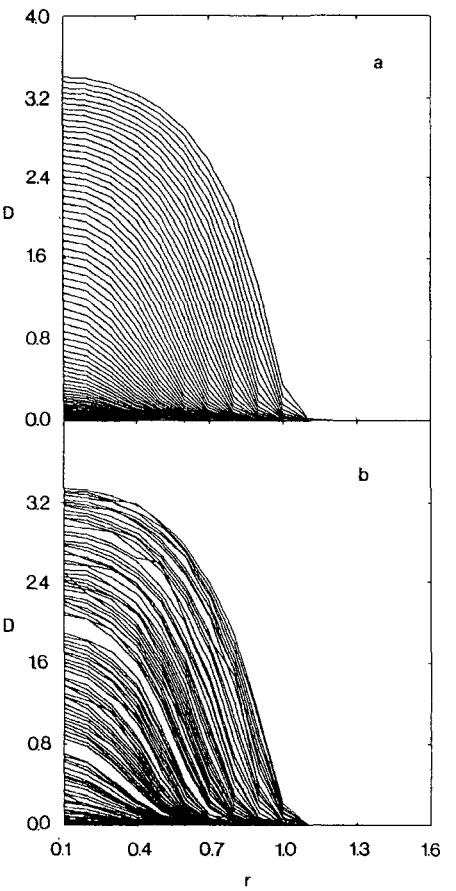

Fig. 2. Model density profiles $D(r)$: (a) without noise; (b) with Poisson noise.

Table 1 gives the results for the three different versions of the conjugate gradient method, when applied to the internal calibration problem. At first sight the results do not differ much. All methods arrive at nearly the same average residual of the objective function. The number of iterations for (DAN) is less then the one for both of the other methods. But in every iteration in (DAN) the Hessian of the objective function has to be evaluated. This additional effort nearly compensates, or in the case of noisy data overcompensates the advantage of a small number of iterations. *Figs. 3* and *4* illustrate the results of such a minimization procedure.

For simulated annealing tests have been carried out with varying choices for the three control parameters, starting temperature T_0, number of iterations per fixed temperature N and cooling rate q. The results are shown in Table 2.

Two general conclusions are: the number of function evaluations increases with increasing cooling rate q, and the average residual of the objective function decreases with increasing cooling rate.

The best results were obtained with small cooling rate q and large N (number of iterations per fixed temperature). This result is also supported by theory.

In this case, the results of simulated annealing and the conjugate gradient method arrive at nearly the same average residual of the objective function. Although the number of function evaluations for simulated annealing exceeds that of the deterministic conjugate gradient method, it is of the same order of magnitude. Thus it becomes difficult to decide which algorithm should be preferred. To draw a final conclusion, one might say that simulated annealing positively compares with deterministic optimization algorithms. It gains further attraction when the simplicity of the algorithm is also taken into account.

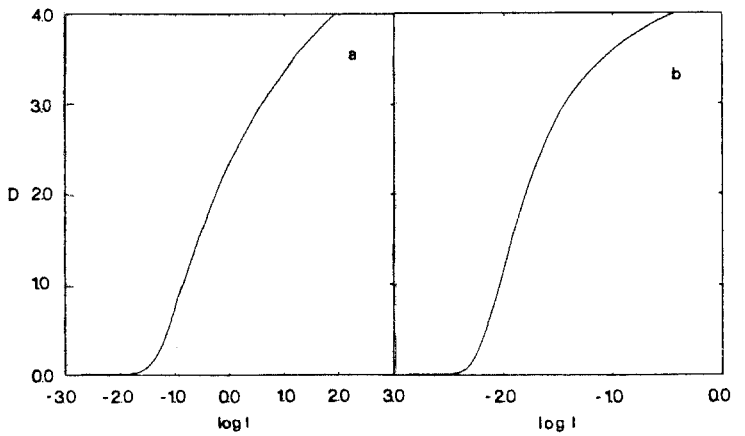

Fig. 3. Characteristic curves: (a) for starting parameters; (b) for solution parameters.

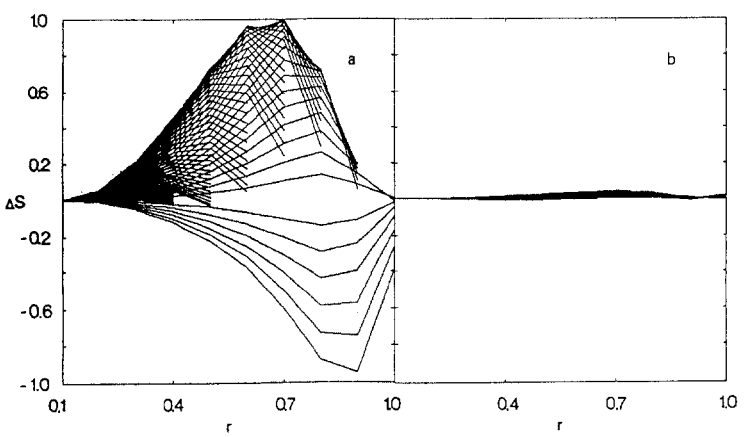

Fig. 4. Differences ΔS between all profiles and a reference profile taken from the same sample: (a) using the characteristic curve with starting parameters; (b) with solution parameters (see *Fig. 3*).

Table 2. Results for simulated annealing

a) model data without noise

T_0	N	q	# f	min. res.	max. res.	aver. res.
100	20	0.1	187	0.01	1.18	0.24
100	20	0.3	184	0.04	1.83	0.24
100	20	0.5	277	0.02	2.11	0.22
100	20	0.7	380	0.03	6.33	0.31(0.06*)
100	20	0.9	410	0.03	2.08	0.24
30	40	0.1	315	0.01	0.13	0.05
30	40	0.3	290	0.03	0.51	0.06

b) model data with Poisson noise

T_0	N	q	# f	min. res.	max. res.	aver. res.
100	20	0.1	141	0.40	10.08	0.24
100	20	0.3	158	0.40	20.86	2.24
100	20	0.5	188	0.40	4.45	0.91
100	20	0.7	244	0.39	3.15	0.63
100	20	0.9	411	0.48	7.20	0.76(0.49*)
30	40	0.1	221	0.40	4.52	0.67(0.47*)
30	40	0.3	240	0.38	4.49	0.69(0.51*)

* without maximal residual

References

Bunclark, P., Irwin, M.J., 1983. In *Statistical Methods in Astronomy*, ESA SP-201, p. 195.

Cèrny, V., 1985. *J. Optim. Theory and Appl.*, **45**, 41.

Hömberg, D., 1988. In *IAU Workshop Astrophotography*, ed. Marx, S., Springer-Verlag, Berlin, p. 77.

Honeycutt, R.K., Chaldu, R.S., 1970. *AAS Photo-Bull.*, **2**, 14.

Horstmann, H., 1988. *These proceedings*, p. 111.

Jeffrey, W., Rosner, R., 1986. *Astrophys. J.*, **310**, 473.

Kirkpatrick, S., Gelatt Jr., S.D., Vecchi, M.P., 1983. *Science*, **220**, 671.

Laarhoven, P.J.M. van, Aarts, E.H.L., 1987. *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht.

Luenberger, D.G., 1972. *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA.