A Multi–Parameter Method for Nonlinear Least–Squares Approximation

R. Schaback

Abstract. For discrete nonlinear least-squares approximation problems $\sum_{j=1}^{m} f_j^2(x) \rightarrow \min$ for m smooth functions $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$ a numerical method is proposed which first minimizes each f_j separately and then applies a penalty strategy to gradually force the different minimizers to coalesce. Though the auxiliary nonlinear least-squares method works on $\mathbb{R}^{n \cdot m}$, it is shown that the additional computational requirements consist of $m - 1$ gradient evaluations plus $\mathcal{O}(m \cdot n)$ operations. The application to discrete rational approximation is discussed and numerical examples are given.

§1. Introduction

Assume a nonlinear unconstrained mimimization problem

$$f(x) \rightarrow \min_{x \in \mathbb{R}^n} ! \tag{1}$$

for a smooth function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ to be given, and assume the problem (1) to have several local minimizers. We want to construct an algorithm that enhances the probability to find global minimizers. Led by the application to nonlinear least squares problems (which will be described in section 5) we assume f to be additively decomposable in the form

$$f(x) = \sum_{j=1}^{m} f_j(x), \qquad f_j : \mathbb{R}^n \rightarrow \mathbb{R}^n \;\; \text{smooth} \tag{2}$$

where each f_j is easier to minimize than f itself.

Mathematical Methods in CAGD and Image Processing
Tom Lyche and L. L. Schumaker (eds.), pp. 1–20.

ISBN 0-12-000000-X.

The basic idea of the multi–parameter algorithm to be constructed will be borrowed from multiple shooting methods in ordinary differential equations: each f_j gets a separate parameter vector $y_j \in \mathbb{R}^n$ to be optimized and then the separate minimizing y_j will be gradually forced to coalesce, joining the separate minimizers gradually into a global solution. This strategy is not confined to additive decompositions. In more detail, we proceed along the following steps:

1. Starting from a given vector $x^{(0)} \in \mathbb{R}^n$, set $y_j^{(0)} := x^{(0)}$, $1 \le j \le m$ and calculate a (possibly only local) minimizer $y_j^{(1)} \in \mathbb{R}^n$ of f_j for $1 \le j \le m$.
2. Choosing a nonnegative penalty function P with control parameters Λ to be described later, perform an iterative minimization of the function

$$\sum_{j=1}^{m} f_j(y_j + y) + P(y_1, \dots, y_m; \Lambda), \qquad y, y_j \in \mathbb{R}^n,\ 1 \le j \le m \tag{3}$$

   on $(y_1, \dots, y_m, y) \in \mathbb{R}^{n\cdot(m+1)}$, starting in $(y_j^{(1)}, \dots, y_j^{(1)}, 0)$ and producing a minimizer $(y_1^{(2)}, \dots, y_m^{(2)}, y^{(2)})$. Here we assume $P(y_1, \dots, y_m, \Lambda) = 0$ iff $y_j = 0$ for all j, and the penalty parameters Λ should be suitably controlled during the optimization.
3. Starting from $x^{(2)} := y^{(2)} + \frac{1}{m} \sum\limits_{j=1}^{m} y_j^{(2)}$, minimize (1) on $\mathbb{R}^n$ by an arbitrary unconstrained optimization procedure to get a minimizer $x^{(3)} \in \mathbb{R}^n$.

Note that the algorithm consists of a single execution of steps 1–3. However, steps 2 and 3 contain an inner iteration.

Depending on the penalty strategy in step 2, we will often have $y_j^{(2)} = 0$ for all j and $x^{(2)} = y^{(2)}$ will already be a local minimizer of f, such that step 3 will often not be necessary at all. Nevertheless, the overall algorithm works from $\mathbb{R}^n$ to $\mathbb{R}^n$ as $x^{(0)} \mapsto x^{(3)}$ and the detour via $\mathbb{R}^{n\cdot(m+1)}$ in step 2 can be concealed from the user.

We could do step 2 without the additional vector y and minimize

$$\sum_{j=1}^{m} f_j(y_j) + P(y_1 - \overline{y}, \dots, y_m - \overline{y}; \Lambda)$$

for $(y_1, \dots, y_m) \in \mathbb{R}^{m\cdot n}$, where $\overline{y} := \frac{1}{m} \sum_{j=1}^{m} y_j$. However, the form used in (3) is more easily handled.

The main problems to be addressed in the sequel are

1. Can the computational effort be reduced to a reasonable amount?
2. How should Λ in step 2 be controlled?
3. Are there some classes of problems where the algorithm is superior over conventional methods?

For simplicity, we shall confine the major part of the rest of the paper to nonlinear least–squares calculations and postpone the general nonlinear optimization problem to later investigations.

§2. Penalty functions

Depending on how much control is required, one can consider different penalty functions of the general form $P(y_1, \dots, y_m; \Lambda)$, e.g.:

$$\lambda^2 \sum_{j=1}^{m} \|y_j\|_2^2, \quad \lambda \in \mathbb{R}, \text{ or} \tag{4}$$

$$\sum_{j=1}^{m} \lambda_j^2 \|y_j\|_2^2, \quad \lambda_j \in \mathbb{R},\ 1 \le j \le m, \text{ or} \tag{5}$$

$$\sum_{j=1}^{m} \|\Lambda_j y_j\|_2^2 \tag{6}$$

with $n \times n$ nonsingular diagonal matrices

$$\Lambda_j = \begin{pmatrix} \lambda_{j1} & & 0 \\ & \ddots & \\ 0 & & \lambda_{jn} \end{pmatrix}, \quad 1 \le j \le m. \tag{7}$$

We shall see later that good control of the parameters of the penalty function is a major problem which forces us to restrict ourselves to the simple case (4) when specific control strategies are treated later. However, the computational complexity investigations of the next section allow rather general controls of type (5) or (6).

§3. Computational Complexity

We consider a Gauss–Newton minimization step for the general composite objective function

$$\sum_{j=1}^{m} f_j^2(y_j + y) + \sum_{j=1}^{m} \|\Lambda_j y_j\|_2^2, \quad y, y_j \in \mathbb{R}^n,\ 1 \le j \le m \tag{8}$$

with (6) and (7). This arises within step 2 of the algorithm and involves $n \cdot (m+1)$ variables instead of just n variables for the classical approach. Thus we compare it with the corresponding step for minimization of the least–squares objective function

$$f(x) = \sum_{j=1}^{m} f_j^2(x), \quad x \in \mathbb{R}^n \tag{9}$$

to find that there is not too much additional work to be done:

Theorem 1. *A linear multi–parameter least–squares problem arising from linearization of the objective function (8) on the space $\mathbb{R}^{n\cdot(m+1)}$ can be solved by adding $\mathcal{O}(m\cdot n)$ operations to the solution of the corresponding problem arising from linearization of the least–squares function (9) on $\mathbb{R}^n$, which needs $\mathcal{O}(m\cdot n^2)$ operations. However, the number of gradient and function evaluations increases from 1 to m.*

Proof: We can write (8) as

$$\|H_\Lambda(y_1,\ldots,y_m,y)\|_2^2$$

with

$$H_\Lambda(y_1,\ldots,y_m,y) := \begin{pmatrix} \Lambda_1 y_1 \\ \vdots \\ \Lambda_m y_m \\ f_1(y_1+y) \\ \vdots \\ f_m(y_m+y) \end{pmatrix}, \qquad H_\Lambda : \mathbb{R}^{mn+n} \to \mathbb{R}^{mn+m}.$$

Writing gradients as row vectors, we calculate the Jacobian of H_Λ as

$$J_\Lambda = \begin{pmatrix} \Lambda_1 & 0 & \ldots & 0 & 0 \\ 0 & \Lambda_2 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \ldots & \Lambda_m & 0 \\ \nabla f_1 & 0 & \ldots & 0 & \nabla f_1 \\ 0 & \nabla f_2 & \ldots & 0 & \nabla f_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \ldots & \nabla f_m & \nabla f_m \end{pmatrix}$$

where the argument of ∇f_j is $y_j + y$ throughout. This requires m evaluations of gradients.

We linearize H_Λ at $y_1,\ldots,y_m,y$ and consider the problem

$$\left\| J_\Lambda \begin{pmatrix} z_1 \\ \vdots \\ z_m \\ z \end{pmatrix} + H_\Lambda(y_1,\ldots,y_m,y)\right\|_2^2 \to \min!$$

for $z, z_j \in \mathbb{R}^n$, $1 \le j \le m$. This linear least–squares problem splits into the sum

$$\sum_{j=1}^m \left\| \begin{pmatrix} \Lambda_j \\ \nabla f_j \end{pmatrix} z_j + \begin{pmatrix} \Lambda_j y_j \\ f_j + \nabla f_j \cdot z \end{pmatrix} \right\|_2^2 \tag{10}$$

and we first minimize each single term with respect to z_j for arbitrary fixed z. With a positive definite $n \times n$ diagonal matrix D, vectors $u, v \in \mathbb{R}^n$, and a scalar α we want to find the minimizer $\tilde{x} \in \mathbb{R}^n$ of an expression of the form

$$\left\| \begin{pmatrix} D \\ u^T \end{pmatrix} x + \begin{pmatrix} v \\ \alpha \end{pmatrix} \right\|_2^2 \to \min_x !$$

A straightforward calculation yields

$$\tilde{x} = -(D^2 + uu^T)^{-1}(Dv + \alpha u) = -D^{-1}v - \gamma D^{-2}u$$

with

$$\begin{aligned} \gamma &= \frac{\alpha - u^T D^{-1} v}{1 + u^T D^{-2} u}, \\ \left\| \begin{pmatrix} D \\ u^T \end{pmatrix} \tilde{x} + \begin{pmatrix} v \\ \alpha \end{pmatrix} \right\|_2^2 &= \gamma^2 (1 + \|D^{-1}u\|_2^2) \\ &= (\alpha - u^T D^{-1} v)^2 / (1 + \|D^{-1}u\|_2^2). \end{aligned}$$

Thus the minimizer of (10) for fixed z satisfies

$$z_j = -y_j - \frac{f_j - \nabla f_j \cdot (y_j - z)}{1 + \|\Lambda_j^{-1} \nabla f_j\|_2^2} \, \Lambda_j^{-2} (\nabla f_j)^T \tag{11}$$

as a function of z and we insert (11) into (10) to get another linear least–squares problem for the vector $z \in \mathbb{R}^n$ as

$$\sum_{j=1}^{m} \frac{1}{1 + \|\Lambda_j^{-1} f_j\|_2^2} \left(f_j - \nabla f_j \cdot (y_j - z)\right)^2 \to \min_z \tag{12}$$

which can be considered as a variation of a usual m by n least–squares calculation required after linearization of (2). The additional work needed for (12) arises in forming the m by n coefficient matrix and the m right–hand–side entries, and is of order $\mathcal{O}(m \cdot n)$. After solving (12) by orthogonalization or singular–value decomposition, the rest will only consist of substitution of z into (11), which again will introduce $\mathcal{O}(m \cdot n)$ operations. ■

For the penalty function (4) and a Levenberg–Marquardt strategy based on normal equations a similar result was obtained by K. Nottbohm in [4].

§4. Control of penalty parameters

Discrete least–squares approximation of a function $\varphi : T \to \mathbb{R}$ on a finite set $T = \{t_1, \ldots, t_m\}$ of m distinct points by a parametrized family of functions

$$\Phi(x, \cdot) : T \to \mathbb{R}, \quad x \in \mathbb{R}^n$$

takes the form (9) with

$$f_j(x) = \Phi(x, t_j) - \varphi(t_j), \quad 1 \leq j \leq m. \qquad (zwoelfa)$$

Section 5 will treat rational approximation as an example for this approach. In most applications the sets

$$X_j := \{x \in \mathbb{R}^n \mid \Phi(x, t_j) = \varphi(t_j)\}, \quad 1 \leq j \leq m$$

are non–empty and easy to reach by minimization of $f_j^2(x)$ on $\mathbb{R}^n$. Consequently, the direct separate minimization of each f_j^2 in step 1 will normally produce parameters $y_j^{(1)} \in X_j$. In most cases the interpolation equation

$$f_j(y_j^{(1)}) = \Phi(y_j^{(1)}, t_j) - \varphi(t_j) = 0 \qquad (14)$$

still has $n-1$ degrees of freedom available for variation of $y_j^{(1)}$. This freedom should be used to find parameters $y_j^{(1)}$ which are "as equal as possible" without violating the interpolation condition (14).

Thus it is advisable to begin step 2 of the algorithm by keeping all penalty parameters at very small positive and fixed values and to iterate on H_Λ within step 2 until no reasonable progress is possible.

The overall control strategy can be illustrated by a diagram like Figure 1, where the values $f = \sum_{j=1}^{m} f_j^2(y_j + y)$ of iterates are plotted against g, where g is the value of the penalty function (4) for $\lambda = 1$.

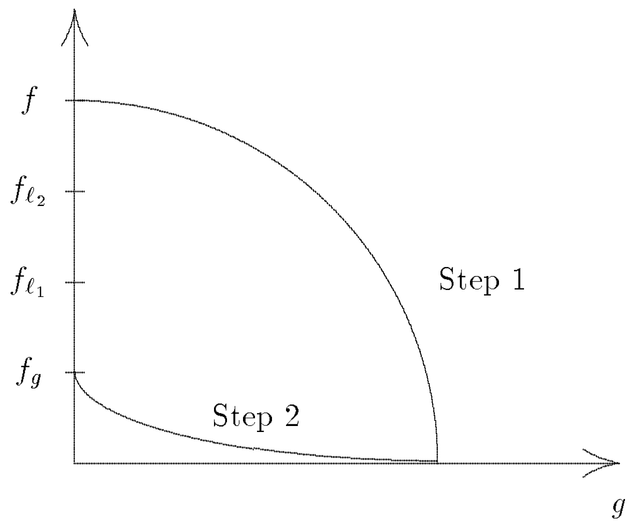

Figure 1. The f - g diagram

The global minimizer of $f = \sum_{j=1}^{m} f_j^2(x)$ on $\mathbb{R}^n$ yields a minimal value f_g on the f axis, while the values of local best approximations are denoted by $f_{\ell_1}, f_{\ell_2}, \ldots$ in Figure 1. A typical application of step 1 will dash down to the g axis from a high starting point on the f axis. Then our suggestion to start step 2 with small values of penalty parameters will cause the method to creep along the g axis towards the origin, avoiding large values of f. This can be put on a more rigorous basis by using small values of ρ in

Theorem 2. *Let the given starting values* $y_j := x^{(1)},\ 1 \le j \le m,\ y := 0$ *and* $\rho > 0$ *satisfy*

$$\rho > \sum_{j=1}^{m} f_j^2(x^{(1)}).$$

The minimization of

$$\sum_{j=1}^{m} f_j^2(y_j + y) + \lambda^2 \sum_{j=1}^{m} \|y_j\|^2 \tag{15}$$

for the penalty function (4) with λ *bounded by*

$$0 < \lambda^2 \le \frac{\rho - \sum_{j=1}^{m} f_j^2(x^{(1)})}{m\|x^{(1)}\|^2}$$

will yield parameters $y^{(2)}, y_j^{(2)}$ $1 \le j \le m$ *with*

$$\sum_{j=1}^{m} f_j^2(y_j^{(2)} + y^{(2)}) \le \rho,$$

provided that an iteration with guaranteed descent of (15) is used.

Proof: For each iterate,

$$\begin{aligned}
\sum_{j=1}^{m} f_j^2(y_j + y) &\le \sum_{j=1}^{m} f_j^2(y_j + y) + \lambda^2 \sum_{j=1}^{m} \|y_j\|_2^2 \\
&\le \sum_{j=1}^{m} f_j^2(x^{(1)}) + \lambda^2 \sum_{j=1}^{m} \|x^{(1)}\|_2^2 \\
&\le \rho. \qquad \blacksquare
\end{aligned}$$

Similar results can easily be obtained for the other penalty functions. The major purpose of Theorem 2 is to keep the principal error term within reasonable bounds while trying to coalesce the parameters as far as possible.

Another easy result follows for large λ in the penalty function (4):

Theorem 3. *Assume* y_j^μ, y^μ *and* $y_j^\lambda, y^\lambda,\ 1 \le j \le m$ *in* $\mathbb{R}^n$ *satisfy*

$$\begin{aligned}
\sum_{j=1}^{m} f_j^2(y_j^\mu + y^\mu) + \mu^2 \sum_{j=1}^{m} \|y_j^\mu\|_2^2 &\le \sum_{j=1}^{m} f_j^2(y_j^\lambda + y^\lambda) + \mu^2 \sum_{j=1}^{m} \|y_j^\lambda\|_2^2 \\
\sum_{j=1}^{m} f_j^2(y_j^\lambda + y^\lambda) + \lambda^2 \sum_{j=1}^{m} \|y_j^\lambda\|_2^2 &\le \sum_{j=1}^{m} f_j^2(y_j^\mu + y^\mu) + \lambda^2 \sum_{j=1}^{m} \|y_j^\mu\|_2^2.
\end{aligned} \tag{16}$$

Then for $\mu < \lambda$ we have

$$\begin{aligned} \sum_{j=1}^m \|y_j^\lambda\|_2^2 &\le \sum_{j=1}^m \|y_j^\mu\|_2^2, \\ \sum_{j=1}^m f_j^2(y_j^\mu + y^\mu) &\le \sum_{j=1}^m f_j^2(y_j^\lambda + y^\lambda). \end{aligned} \tag{17}$$

If the monotonic function

$$\sum_{j=1}^m f_j^2(y_j^\lambda + y^\lambda) + \lambda^2 \sum_{j=1}^m \|y_j^\lambda\|_2^2$$

stays bounded for $\lambda \to \infty$, then $y_j^\lambda \to 0$ for $\lambda \to \infty$ and all j, $1 \le j \le m$.

The theorem is a consequence of the theory of penalty function methods (see e.g.: Fletcher [2]). ■

Inequalities (16) will typically hold if (15) is minimized by a method that ensures (weak) descent and if y_j^μ, y^μ and y_j^λ, y^λ are optimal for minimization for μ and λ fixed, respectively, where y_j^μ, y^μ are admissible for minimization with fixed λ, and conversely. Then Theorem 3 allows to force the y_j^λ to coalesce for large λ, if the penalty function (15) remains bounded during the iteration. This can be used as the final strategy of the internal iteration within step 2 of the basic algorithm. Standard arguments of nonlinear optimization show that minimization of (15) for large λ is equivalent to minimization of (2) under a rather strict constraint

$$\sum_{j=1}^m \|y_j\|^2 \le \varepsilon$$

on the y_j. Thus, if the limits

$$\lim_{\lambda\to\infty} y_j^\lambda = 0, \qquad \lim_{\lambda\to\infty} y^\lambda = y$$

exist, the point y is critical for (2). This shows that step 3 is not necessary, if the penalized objective function remains bounded when the penalty parameters are driven to infinity in step 2.

The two extreme cases described above leave the strategy of the internal iteration within step 2 for "moderate" values of λ open. As described by (17), pushing λ up will normally

$$\begin{aligned} \text{increase} \quad \sum_{j=1}^m f_j^2(y_j^\lambda + y^\lambda) \quad &=: \quad f^\lambda, \quad \text{and} \\ \text{decrease} \quad \sum_{j=1}^m \|y_j^\lambda\|_2^2 \quad &=: \quad g^\lambda, \end{aligned}$$

while lowering λ has the opposite effect. If the value of f^λ gets much larger as expected or wanted when pushing up λ, the user may prefer to go back to $\lambda = 0$ for a couple of iterations, hoping that this restart will give a better starting point for step 2. Monitoring the values (f^λ, g^λ) in a diagram like Figure 1 may help to detect progress.

For all penalty functions of section 2 the critical points $(y_1^\Lambda, \ldots, y_m^\Lambda, y^\Lambda)$ of $\|H_\Lambda\|^2$ for Λ fixed correspond to points $z_j^\Lambda = y_j^\Lambda + y^\Lambda$ of the set

$$M := \left\{ \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} \in \mathbb{R}^{mn} \mid \sum_{j=1}^m f_j(z_j) \cdot \nabla f_j(z_j) = 0 \right\}.$$

This follows from

$$\frac{1}{2}\nabla_y \|H_\Lambda\|^2 = \frac{1}{2}\nabla_y \sum_{j=1}^m f_j^2(y_j + y) = \sum_{j=1}^m (f_j \cdot \nabla_y f_j)(y_j + y)$$

for arguments $(y_1, \ldots, y_m, y)$ of H_Λ, independent of the penalty parameters. Control of Λ will steer $y_j^\Lambda + y^\Lambda$ along M, possibly jumping between connected components of M.

Penalty function (4), if fully optimized for each value of λ, will generically produce pieces of curves on M, which may be visualized as curves in Figure 1 if we plot (f^λ, g^λ) as a function of λ. However, we do not recommend to follow these curves closely, because this would spoil the main advantage of the method: its high space dimension allows a lot of freedom to manoeuver around. Performing just one linearization step for each instance of penalty parameters Λ is quite enough and allows much more freedom, jumping between many trajectories on M.

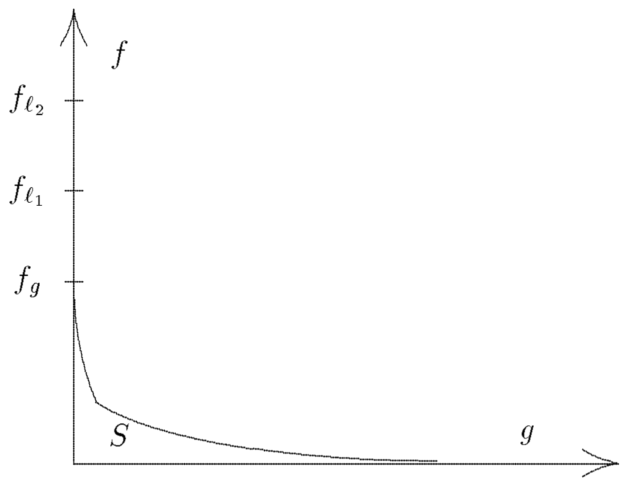

Figure 2. The unattainable set S

Theoretically, the best strategy would be to follow the boundary of the "unattainable set"

$$S = \left\{ (f,g) \in \mathbb{R}^2_{\geq 0} \;\middle|\; \begin{array}{l} \text{if } g = \sum\limits_{j=1}^{m} \|y_j\|^2,\ y_j \in \mathbb{R}^n, \text{ then} \\ \sum\limits_{j=1}^{m} f_j^2(y_j + y) > f \text{ for all } y \in \mathbb{R}^n \end{array} \right\}$$

in Figure 2, which always links the global minimum of (2) with the global minimum of $\sum_{j=1}^m f_j^2(y_j)$ having least value of $\sum\limits_{j=1}^{m} \|y_j\|^2$. However, this is not easier to solve than the original problem, but it indicates that one should try to keep as "southwest" in Figures 1 and 2 as possible.

Since the set M is described by n equations on $\mathbb{R}^{n\cdot m}$, the penalty function (6) has enough degrees of freedom to allow appropriate "steering" along M in a sophisticated way that may involve a lot of information about $f_j(y_j)$ or $\nabla f_j(y_j)$. Thus the user may assign large penalty parameters to "good" y_j (or components thereof) to make sure that they will not be moved around too much. Details can easily be studied for $m = 2$ and $n = 1$, but we will leave further analysis to future work.

Ute Jäger [3] treated the penalty function (4) with $\lambda^2 \sim (g^\lambda)^{-1}$ and showed that nondegenerate attractors $\tilde{x}$ for (9) on $\mathbb{R}^n$ yield nondegenerate attractors $(0,\ldots,0,\tilde{x})$ for (15) on $\mathbb{R}^{n(m+1)}$, provided that a well–controlled trust–region method is used.

§5. Application to discrete rational approximation

On a finite subset $T = \{t_1,\ldots,t_m\} \subset [-1,+1]$ we consider the approximation of a function $\varphi : T \to \mathbb{R}$ by rational functions

$$\Phi(x,t) = \frac{p(x,t)}{q(x,t)} \qquad t \in [-1,+1] \tag{18}$$

where

$$p(x,t) = \sum_{i=1}^{k} x_i t^{i-1}, \quad q(x,t) = 1 + \sum_{i=k+1}^{n} x_i t^{i-k}.$$

We use (13) and obtain a nonlinear least–squares problem of type (9).

To study step 1 of the algorithm, we note that $f_j(x)$ vanishes on the affine subspace of $\mathbb{R}^n$ consisting of all $x \in \mathbb{R}^n$ with

$$p(x,t_j) - \varphi(t_j)\cdot q(x,t_j) = 0 \tag{19}$$

for $1 \leq j \leq m$ except for points where

$$p(x,t_j) = 0 = q(x,t_j)$$

holds, and at which l'Hospital's rule for evaluating $\Phi(x, t_j)$ does not yield $\varphi(t_j)$. If we ignore these exceptional x for a moment, we find that minimizers $y_j^{(1)}$ of f_j^2 with $f_j(y_j^{(1)}) = 0$ exist and form an affine subspace. Thus there normally are no problems with step 1 of the algorithm.

If step 2 is carried out with very small penalty parameters, it is comparable to a minimization of the positive definite penalty term under the affine constraints (19). Thus step 2 yields a unique minimizer $(y_1^{(2)}, \ldots, y_m^{(2)}, y^{(2)})$ depending on Λ, but not on common factors of the various possible penalty variables in Λ. In particular, the minimizer does not depend on the result of step 1. It is a multiple parameter set that satisfies all linearized one–point interpolation problems

$$p(y_j^{(2)}, t_j) - \varphi(t_j)q(y_j^{(2)}, t_j) = 0,\ 1 \le j \le m,$$

and step 2 has tried to make the parameters $y_j^{(2)}$ as equal as possible. Thus for all discrete rational approximation problems a fixed implementation of the multi–parameter method will almost surely produce the same output for each possible starting parameter vector, since the exceptional points form a set of measure zero. Of course, this feature does not generalize to other nonlinear families of approximating functions.

§6. Examples

We take the example

$$f^2(x, y) = \sum_{i=1}^{m} f_i^2(x, y), \quad \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \\ f_3(x,y) \\ f_4(x,y) \end{pmatrix} = \begin{pmatrix} 1 - x + 25xy \\ 1 + x \\ 1 - y \\ 1 + y \end{pmatrix}$$

from [4] for $n = 2,\ m = 4$ with three critical points:

$$\begin{array}{rl} \text{at } (x,y) = (0,0) & \text{is a saddle point with } f(x,y) = 2 \\ \text{at } (x,y) = (0.12, -0.24) & \text{is a local minimum with } f(x,y) = 1.843 \\ \text{at } (x,y) = (-1.006, 0.079) & \text{is a global minimum with } f(x,y) = 1.419. \end{array}$$

For various possibilities of controlling λ in step 3 we always found the multi–parameter method to converge to the global minimum (see also [3]). Figure 3 contains a contour plot of the approximation error for this example, where the saddle point and the global minimum are clearly visible. The local minimum is overlaid by its surrounding (black) basin of attraction of a regularized Gauss–Newton method. Regularization was done by a Levenberg–Marquardt strategy to prevent rank loss, and damping by a stepsize strategy was used to enforce descent even for small or zero Levenberg–Marquardt parameters. The same Gauss–Newton routine was used in step 3 of the multi–paremeter algorithm. The corresponding plot for the multi–parameter method does not contain the black blob, because the basin of attraction of the local minimum with respect to the multi–parameter method was empty.

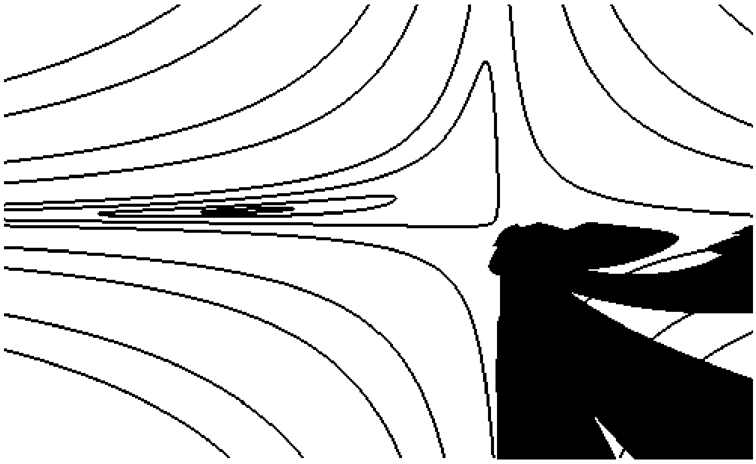

Figure 3. Contours and attractors

The next example is taken from [1] and approximates $\varphi(t) = t^2 - 0.6$ on 11 equidistant points of $[-1, 1]$ by rational functions (18) with constant numerator and linear denominator. The problem is symmetric, and we plot the error in the (x_1, x_2)–plane in Figure 4. Note that there are singularities for $x_2 = -5/j$ for $1 \leq |j| \leq 5$, and the horizontal axis $x_2 = 0$ extends through the global minimum at $x_1 = -0.2$, $x_2 = 0$ near the lower margin of Figure 4. The local minima and error values ϵ are

$$\begin{array}{lllllll}
x_1 & = & -0.200000 & x_2 & = & 0.000000 & \epsilon = 1.172 \\
x_1 & = & -0.035051 & x_2 & = & \pm 1.075847 & \epsilon = 1.232 \\
x_1 & = & -0.053199 & x_2 & = & \pm 1.459828 & \epsilon = 1.219 \\
x_1 & = & -0.062364 & x_2 & = & \pm 2.204753 & \epsilon = 1.211 \\
x_1 & = & -0.067159 & x_2 & = & \pm 4.423534 & \epsilon = 1.207
\end{array}$$

the latter differing only by about 3 %, making the problem hard to solve globally, if not started near the global minimum. Figure 5 shows the basins of attraction of the stabilized Gauss–Newton method, where we used black and white alternatively for the attractors.

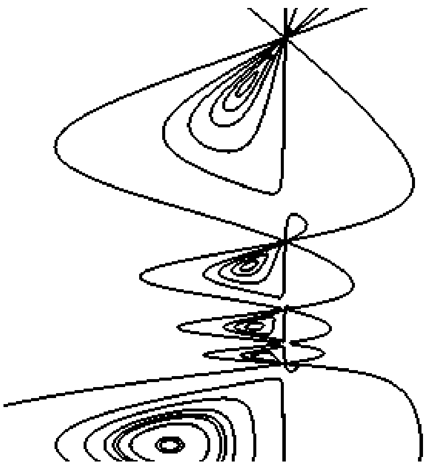

Figure 4. Contours of approximation error

The multi–parameter method always converged to the global minimum, though the global minimum does not have the "nicest" basin of attraction. Other discrete rational approximation problems we tested were less hazardous. In all cases the results were qualitatively the same. Tests on discrete exponential approximation problems, however, did not show such a good behaviour.

Figure 5. Gauss–Newton attractors

Acknowledgements:

Special thanks go to Dr. Immo Diener for a series of useful discussions and some valuable help preparing the final version of the paper.

References

1. Diener, I., On Nonuniqueness in Nonlinear L_2-Approximation, J. Approx. Th. 51 (1987), 54-67
2. Fletcher, R., Practical Methods of Optimization Vol.1: Unconstrained Optimization, Wiley, Chichester, 1980
3. Jäger, U., Das Mehrfach–Parameter–Verfahren unter Benutzung der trust region–Methode zur Lösung nichtlinearer, diskreter Approximationsprobleme in der L_2–Norm, Diplomarbeit, Göttingen 1989.
4. Nottbohm, K., Das Mehrfach–Parameter–Verfahren zur Lösung nichtlinearer diskreter Approximationsprobleme in der L_2–Norm. Ph.D. Dissertation, Göttingen 1986.

R. Schaback
Georg–August–Universität Göttingen
Lotzestrasse 16–18
D–3400 Göttingen, Germany
schaback@ namu01.gwdg.de