An Improved Multi-Parameter Adjustment Algorithm for Inverse Eigenvalue Problems

B. Hofmann


Предлагается и мотивируется алгоритм обработки проблем многопараметрической регуляризации. На основе вычислительных экспериментов при одной обратной задачи на собственные значения возможно найти улучшенная версия алгоритма.

A multi-parameter adjustment algorithm for regularization is established and motivated. Based on numerical experiments concerning the solution of an inverse eigenvalue problem an improved version of this algorithm can be suggested.

1. Introduction

We are going to consider a discretized inverse problem

$$Ax = z \quad (x \in D \subseteq \mathbb{R}^n, \ z \in \mathbb{R}^m) \quad (1)$$

(cf. [2, Chap. 3]). The operator $A: D \rightarrow \mathbb{R}^m$ will be continuous and in general nonlinear. Moreover, the set $D$ of admissible solutions $x$ of problem (1) is assumed to be closed and convex. Let $A = (A_1, A_2, \ldots, A_k)$ and $z = (z_1, z_2, \ldots, z_k)$ denote a pair of decompositions of the operator $A$ describing the direct problem and of the observation data vector $z$, respectively, where $A_i: D \rightarrow \mathbb{R}^{m_i}$ are also continuous operators and $z_i \in \mathbb{R}^{m_i}$ ($i = 1, 2, \ldots, k$; $m_1 + \cdots + m_k = m$) the associated observation sub-vectors. We suppose the discretization error of problem (1) to be negligible in comparison with the observation errors $\delta_i > 0$ according to the data $z_i$ ($i = 1, 2, \ldots, k$). The model under consideration also applies to control problems. Then, $z$ is the vector of desired values which are to be achieved by an appropriate choice of the control vector $x \in D$.

Using the discrepancy principle (cf. e.g. [2, Chap. 4]) for obtaining a regularized solution to the finite-dimensional problem (1), the optimization problem

$$\min_{x \in X_0} \Omega(x) \quad (2)$$

with

$$X_0 = \{ x \in D: \| A_i x - z_i \| \leq \delta_i \quad (i = 1, 2, \ldots, k) \}$$

has to be solved. Here, $\Omega$ designates a nonnegative continuous stabilizing functional on $\mathbb{R}^n$, i.e., the level sets $\{ x \in D: \Omega(x) \leq c \}$ are compact or empty for all $c \geq 0$. Furthermore, $\| \cdot \|$ is the Euclidean vector norm in finite-dimensional spaces. We assume $X_0 \neq \emptyset$. Then there is at least one minimizer $x_{\text{opt}}$ of $\Omega$ subject to $X_0$. One can show that the non-empty set $X_{\text{opt}}$ of solutions to problem (2) stably depends on the data $z$. 
Note that for nonlinear operators $A$, a minimizer $x_{\text{opt}}$ of $Q$ subject to $X$ is difficult to compute in a straightforward manner. This is due to the fact that the domain $X$ frequently fails to be convex and connected. For such optimization problems no advanced direct numerical methods are available. Therefore, we consider the class of multi-parameter auxiliary problems

$$\min_{x \in D} F(x, \lambda) \quad (\lambda \in \mathbb{R}^k)$$

with

$$\mathbb{R}_{+}^k = \{ \lambda \in \mathbb{R}^k : \lambda_1, \ldots, \lambda_k \geq 0 \}.$$

$$F(x, \lambda) = \sum_{i=1}^{k} \lambda_i \| A_i x - z_i \|^2 + \Omega(x) \quad (\lambda \in \mathbb{R}_{+}^k).$$

The well-posedness (existence and stability of solutions) of problem (2) carries over to problem (3) since $Q$ has stabilizing character. Thus at least one minimizer $x_1$ of $F(\cdot, \lambda)$ subject to $D$ exists for all $\lambda \in \mathbb{R}_{+}^k$. However, contrary to problem (2) the advantage of solving (3) numerically lies in the convexity of the domain $D$.

2. The multi-parameter adjustment algorithm

In this section, we wish to study the chances of approximating minimizers $x_{\text{opt}}$ according to problem (2) as an iteration process exploiting minimizers $x_i$ of (3) for appropriately chosen parameter vectors $\lambda$. The values $\lambda_i \geq 0$ ($i = 1, 2, \ldots, k$) play the role of Lagrangian multipliers. Obviously,

$$L(x, \lambda) = \sum_{i=1}^{k} \tilde{\lambda}_i (\| A_i x - z_i \|^2 - \delta_i^2) + \Omega(x) \quad (\lambda \in \mathbb{R}_{+}^k)$$

represents the Lagrangian functional of the basic problem (2). If there is a pair of vectors $(\bar{x}, \bar{\lambda}) \in D \times \mathbb{R}_{+}^k$ such that

$$L(\bar{x}, \lambda) \leq L(\bar{x}, \bar{\lambda}) \leq L(x, \lambda) \quad \text{for all } x \in D, \lambda \in \mathbb{R}_{+}^k,$$  

then $(\bar{x}, \bar{\lambda})$ is a saddle point of the Lagrangian saddle point problem according to (2). Hence, $\bar{x} = x_{\bar{\lambda}}$ solves problem (2).

**Lemma:** A couple of vectors $(\bar{x}, \bar{\lambda}) \in D \times \mathbb{R}_{+}^k$ satisfies the inequality (4) if and only if $\bar{x} = x_{\bar{\lambda}}$ is a solution of problem (3), provided that $\lambda = \bar{\lambda}$ fulfills the requirements

$$\tilde{\lambda}_i (\| A_i \bar{x} - z_i \|^2 - \delta_i^2) = 0 \quad (i = 1, 2, \ldots, k)$$

and

$$\| A_i \bar{x} - z_i \| \leq \delta_i \quad \text{if} \quad \tilde{\lambda}_i = 0 \quad (i = 1, 2, \ldots, k).$$

**Proof:** In view of $L(x, \lambda) = F(x, \lambda) - (\lambda_1 \delta_1^2 + \cdots + \lambda_k \delta_k^2)$ the right-hand inequality of (4) is equivalent to $\bar{x} = x_{\bar{\lambda}}$. The left-hand inequality corresponds to the estimation

$$\sum_{i=1}^{k} (\tilde{\lambda}_i - \lambda_i) (\| A_i \bar{x} - z_i \|^2 - \delta_i^2) \geq 0 \quad (\lambda_1, \ldots, \lambda_k \geq 0).$$

This coincides with (5) and (6). Thus the proof is complete.

For an arbitrary chosen value $\mu > 0$ the equation (5) may be rewritten equivalently as

$$\tilde{\lambda}_i (\| A_i \bar{x} - z_i \|^\mu - \delta_i^\mu) = 0 \quad (i = 1, 2, \ldots, k)$$
An Improved Multi-Parameter Adjustment Algorithm

and

$$\lambda_i = \frac{\| A_i \bar{x} - z_i \|}{\epsilon} \quad (i = 1, 2, \ldots, k).$$  \hspace{1cm} (7)

Based on formula (7) we can propose an iteration process that yields vectors \(x^{(j)}\) \(j = 0, 1, 2, \ldots\) approaching \(x_{\text{opt}}\). Let

$$x^{(j)}_{i > 0} = (x_{ji})_{i > 0}, \quad \lambda^{(j)} = (\lambda_1^{(j)}, \ldots, \lambda_k^{(j)}).$$  \hspace{1cm} (8)

be a sequence of solutions to problem (3) according to the parameter vector sequence \((\lambda^{(j)})_{j > 0}\) determined by

$$\lambda_i^{(j+1)} = \lambda_i^{(j)} \max \left( \frac{\| A_i \bar{x}^{(j)} - z_i \|}{\epsilon}, \delta_i \right) \quad \left( i = 1, 2, \ldots, k \right),$$  \hspace{1cm} (9)

where the small value \(\epsilon\) with \(0 < \epsilon \ll 1\) and an initial guess \(\lambda^{(0)}\) with strictly positive components \(\lambda_i^{(0)} > 0\) \((i = 1, 2, \ldots, k)\) are chosen in an appropriate way. For the introduced iteration process we can formulate the following proposition.

**Theorem:** Provided that there exists a pair of vectors \((\bar{x}, \bar{\lambda}) \in D \times \mathbb{R}_k^k\) such that \(\|z^{(j)} - \bar{\lambda}\| \to 0\) and \(\|x^{(j)} - \bar{x}\| \to 0\) as \(j \to \infty\), then this pair satisfies the relations (5) and (6), i.e., \(\bar{x} = x_{\text{opt}}\) is a solution of problem (2).

**Proof:** From \(L(x^{(j)}, \lambda^{(j)}) \leq L(x, \lambda)\) it follows \(L(\bar{x}, \bar{\lambda}) \leq L(x, \bar{\lambda})\) for all \(x \in D\). Owing to formula (9), we obtain the equation

$$\lambda_i^{(j+1)} = \lambda_i^{(j)} \max \left( \frac{\| A_i \bar{x}^{(j)} - z_i \|}{\delta_i}, \epsilon \right) \quad \left( i = 1, 2, \ldots, k \right),$$

i.e., either \(\lambda_i^{(j)} = 0\) or \(\| A_i \bar{x}^{(j)} - z_i \| = \delta_i\) for all \(i = 1, 2, \ldots, k\). This provides equation (5). Whenever \(\lambda_i^{(j)} = 0\), then there is a monotonically non-increasing subsequence of positive numbers \(\lambda_i^{(j)} \to 0\) as \(j \to \infty\). That means, \(\| A_i \bar{x}^{(j)} - z_i \| / \delta_i \leq 1\) for all \(j\). Consequently, \(\| A_i \bar{x} - z_i \| \leq \delta_i\) if \(\lambda_i^{(j)} = 0\). Hence the requirement (6) is satisfied and the theorem is proved.

**Multi-parameter adjustment algorithm (MPAA):**

**Step 1:** Choose \(\mu > 0, 0 < \epsilon \ll 1, \epsilon_1 > 0, \epsilon^{(0)} > 0\) and a maximum number \(j_{\text{max}}\) of iteration steps. Set \(j := 0\).

**Step 2:** Compute \(x^{(j+1)} := x_{ji}\) by solving an optimization problem of the form (3). If \(j = j_{\text{max}}\), then set \(x_{\text{alg}} := x^{(j)}\) and stop, otherwise compute \(\lambda_{i+1}^{(j+1)}\) according to formula (9).

**Step 3:** If \(\|z^{(j+1)} - \bar{\lambda}^{(j)}\| \leq \epsilon_1\), then set \(x_{\text{alg}} := x^{(j)}\) and stop, otherwise set \(j := j + 1\) and return to Step 2.

**Remark 1:** Basic ideas of the algorithm MPAA were already proposed by the author in [2, p. 97] and in [7]. An advantage of the special kind of Lagrange multiplier estimation suggested above is the immediate availability of the right-hand side of formula (9). Thus, the total amount of computational work for the iteration process is sufficiently small and can essentially be reduced to the costs of solving the associated optimization problems (3). Note that a quotient-type correc tor iteration based on the right-hand side of (7) for \(\mu = 1\) is also proposed in [6, p. 149–157] for spline-problems.

**Remark 2:** The introduction of a small value \(\epsilon > 0\) in formula (9) is of great theoretical importance for the convergence of the iteration process towards a solution of problem (2). By introducing such an \(\epsilon > 0\) we exclude a situation characterized by \(\lambda_i^{(j)} = 0\) \(j = j_0 + 1,\)
3. A particular inverse eigenvalue problem

Now we consider an inverse problem of control type which corresponds to the computation of eigenvalues of a quadratic matrix. This inverse eigenvalue problem may be written in the form of (1) and (2) with \( k = m \) and \( m_i = 1 \) \((i = 1, 2, \ldots, k)\). Here, the vector \( x \in D := \mathbb{R}^k \) is to be determined so that a nonnegative continuous stabilizing functional \( Q \) attains its absolute minimum over a set \( X_\delta \) of feasible vectors. A vector \( x \) with nonnegative components belongs to \( X_\delta \) if and only if the symmetric positively semidefinite matrix of dimension \( l \geq k \)

\[
M(x) = x_1M_1 + x_2M_2 + \cdots + x_nM_n
\]

possesses the \( k \) largest eigenvalues \( v_1(x) \geq v_2(x) \geq \cdots \geq v_{\delta}(x) \geq 0 \) such that \( |v_i(x) - z_\delta| \leq \delta \) \((i = 1, 2, \ldots, k)\). In this context, the vector \( z \in \mathbb{R}_+^k \) of desired eigenvalue approximations \( z_1 \geq z_2 \geq \cdots \geq z_k \geq 0 \) and the symmetric positively semidefinite matrices of dimension \( l \), \( M_i \) \((i = 1, 2, \ldots, n)\), are prescribed. The general nonlinear operator \( A: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^k \) (cf. formula (1)) transforms the vector \( x \) of multipliers in (10) into the \( k \) largest eigenvalues \( v_i(x) \) \((i = 1, 2, \ldots, k)\) of the matrix \( M(x) \). Inverse eigenvalue problems regarding matrices with an additive structure (10) are for example examined in [5] (see also [1, 3] or [2, p. 55]). The difficulties of our particular problem are associated with the numerical solution of the optimization problem (2). In order to find minimizers of \( Q \) subject to \( X_\delta \), the algorithm MPAA of Section 2 and the improved version introduced below in Section 4 may be applied. The immediate reduction of the constrained problem (2) to a problem without or with easy constraints were desirable since no Lagrange multipliers had to be estimated after that reduction. However, all numerical experiments which tried to solve auxiliary problems

\[
\text{minimize } \left\{ \beta \sum_{i=1}^k \max \left( (|v_i(x) - z_\delta| - \delta_j)^2, 0 \right) + \Omega(x) \right\}
\]

with a penalty parameter \( \beta > 0 \) have completely failed. The objective function to be minimized in (11) tends to get ill-conditioned whenever \( \beta \) becomes sufficiently large. Thus, minimization routines of Newton or Gauss-Newton type are seldom able to find satisfactory approximations of the wanted absolute minimum.

In comparison with the algorithm MPAA or its refinements the solution of (11) requires a many times higher amount of computational work. By our experience the difficulties mentioned above also occur if a one-parameter family of optimization
problems

\[
\minimize_{x \in \mathbb{R}^n} \left\{ \max_{1 \leq i \leq k} \frac{|v_i(x) - z_i|}{\delta_i} + \alpha \Omega(x) \right\}
\]  

(12)

is considered. Using this approach, the regularization parameter \( \alpha > 0 \) should be controlled by the discrepancy principle such that \( \max_{1 \leq i \leq k} \left| \frac{|v_i(x) - z_i|}{\delta_i} \right| = 1 \) holds at the minimum point of (12). Consequently, from the numerical point of view there seems to be no way of avoiding multi-parameter regularization problems (3) when (2) is to be solved with small costs. Especially, if \( \Omega(x) \) represents a quadratic form with respect to \( x \), problem (3) for given \( \lambda \in \mathbb{R}^k \) may be solved by Levenberg-Marquardt modifications to the Gauss-Newton method in an efficient manner.

4. The improved algorithm

The algorithm MPAA was tested for efficiency in solving the inverse eigenvalue problem described in Section 3. Numerical experience proved that for exponents \( 1 \leq \mu \leq 2 \) the iteration process (8), (9) in general converges. It becomes evident that a constant exponent \( \mu > 0 \) for all residual components throughout the whole iteration is not optimal. If \( \mu = 1 \), then the multipliers \( \lambda_i^{(j)} \) converge to \( \lambda \) monotonically. However, the rate of convergence is rather small. For \( \mu = 2 \) we obtain an oscillating iteration, but the geometric mean of two consecutive iterates \( \lambda^{(j+1)} \) frequently provides a good estimate of \( \lambda \). Whenever \( \mu < 1 \) or \( \mu > 2 \), the slow convergence or oscillation is the more strengthened the smaller or greater, respectively, the value \( \mu \) becomes.

We can summarize that an improvement of MPAA requires the permanent inspection of monotonicity with respect to the multiplier sequences \( \lambda_i^{(j)} \) (cf. formula (9)) for fixed \( \mu \). If there is no monotonicity, i.e., \( (\lambda_i^{(j+1)} - \lambda_i^{(j)}) \times (\lambda_i^{(j)} - \lambda_i^{(j+1)}) < 0 \), then the iteration should be continued with a smaller value \( \mu = \mu_{\text{new}} \) (e.g., \( \mu_{\text{new}} := \max (\mu, \mu/2) \)) and \( \lambda_{i_{\text{new}}} := \sqrt[\mu_{\text{new}}]{\lambda_i^{(j)} \lambda_i^{(j+1)}} \). In the monotonicity case, \( (\lambda_i^{(j+1)} - \lambda_i^{(j)}) (\lambda_i^{(j)} - \lambda_i^{(j-1)}) > 0 \), it seems to be a good heuristic strategy to enlarge the iteration exponent. Thus we replace \( \mu \) by \( \mu_{\text{new}} \) satisfying the equation

\[
\lambda_i^{(j+1)} = \lambda_i^{(j)} \| A_i x_i^{(j)} - z_i \|^2 / \delta_i^{(j+1)},
\]

i.e.,

\[
\mu_{\text{new}} = \ln \left( \frac{\lambda_i^{(j+1)}}{\lambda_i^{(j-1)}} \right) / \ln \left( \frac{\| A_i x_i^{(j)} - z_i \|}{\delta_i} \right).
\]

This would provide an acceleration of the multiplier iteration. Obviously, the answer to the monotonicity question and consequently the exponent \( \mu = \mu_i \) thus always depend on \( i \).

Improved multi-parameter adjustment algorithm (IMPAA):

Step 1 (Initialization): Choose \( \mu_{\text{max}} > 1 \), \( 1 \leq \mu_0 < \mu_{\text{max}} \), \( \epsilon_1 > 0 \), \( \lambda^{(0)} > 0 \) and a maximum number \( j_{\text{max}} \) of iteration steps. Set \( j := 0 \).

Step 2 (First iteration step): Compute \( x^{(0)} := x^{(0)} \) (cf. (3)) and

\[
\lambda_i^{(1)} := \lambda_i^{(0)} \| A_i x^{(0)} - z_i \|^{2 \mu_i / 2} / \delta_i^{(1)} \quad (i = 1, 2, \ldots, k).
\]

Set \( \lambda := \lambda^{(1)} \), \( \mu_i := \mu_0 \) (i = 1, 2, \ldots, k) and \( j := 1 \).
Step 3 (Intermediate step): Compute \( x^n := x_{i+1} \) (cf. (3)) and set \( x^a := x^n \) and stop. Otherwise compute

\[
\lambda^a(i+1) := \lambda^b\|A_i x^a - z_i\|^2/\delta^a(i)
\]

and set \( \lambda^a := \lambda^a(i+1) \). If \( \|x^a - \lambda^a\| \leq \epsilon_1 \), then set \( x_{alg} := x^a \) and stop, otherwise set \( j := j + 1 \).

Step 4 (Predictor step): Compute \( x^{(i)} := x_{i} \) (cf. (3)) and set \( x^b := x^{(i)} \). If \( j = j_{max} \), set \( x_{alg} := x^b \) and stop. Otherwise compute

\[
\mu^a(i+1) := \frac{\lambda^b}{\lambda^c} \|A_i x^b - z_i\|^2/\delta^a(i)
\]

and set \( \mu^a := \mu^a(i+1) \). If \( \|x^b - \mu^a\| \leq \epsilon_1 \), then set \( x_{alg} := x^b \) and stop, otherwise set \( j := j + 1 \).

Step 5 (Corrector step): For \( i := 1 \) do:

If \( (\lambda^c - \lambda^b)(\lambda^b - \lambda^a) < 0 \), then set \( \mu^a := \max(1, \mu^a/2) \) and \( \lambda^a(i+1) := \sqrt{\lambda^b/\lambda^c} \lambda^a \), otherwise set

\[
\mu^a := \min\left(\mu_{max}, \frac{\ln(\lambda^a/\lambda^c)}{\ln(\|A_i x^a - z_i\|/\delta_i)}\right), \quad \lambda^a(i+1) := \lambda^b\|A_i x^b - z_i\|^2/\delta^a(i)
\]

endfor.

Step 6 (Return step): If \( \|x^{(i+1)} - x^{(i)}\| \leq \epsilon_1 \), then set \( x_{alg} := x^{(i)} \) and stop, otherwise set \( \lambda^a := \lambda^{(i+1)} \), \( j := j + 1 \) and return to Step 3.

5. Computational results

We complete the paper with a comparison of the efficiency of the algorithms MPAA versus IMPAA applied to the inverse eigenvalue problem described in Section 3. Assume \( k = n = l = 3 \), \( \Omega(\cdot) = \|\cdot\|^2 \) and

\[
M_1 = \begin{pmatrix} 2 & 0 & 2 \\ 0 & 3 & -1 \\ 2 & -1 & 3 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 4 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad M_3 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}
\]

Moreover, let \( z = (11.359..., 7.0, 2.641...) \) be the vector of eigenvalues \( \nu_i \) \( (i = 1, 2, 3) \) of the matrix \( M = M_1 + M_2 + M_3 \) and \( \delta_1 = 0.2, \delta_2 = 0.4, \delta_3 = 1.0 \). The problem under consideration

\[
\text{minimize } x \in \{x \in \mathbb{R}^3 : \nu_i(x) - z_i \leq \delta_i \ (i = 1, 2, 3)\} ||x||^2
\]

(cf. (10)) has a unique solution \( x_{opt} = (0.945..., 1.175..., 0.715...) \) with \( |\nu_1(x_{opt}) - z_1| = 0.2, |\nu_2(x_{opt}) - z_2| = 0.4 \) and \( |\nu_3(x_{opt}) - z_3| = 0.403... \). There is also a uniquely determined multiplier vector \( \lambda = \lambda = (1.060..., 0.151..., 0) \) such that the solution \( x_i \) of

\[
\text{minimize } \left\{ \frac{3}{2} \sum_{i=1}^3 |\nu_i(x) - z_i|^2 + ||x||^2 \right\}
\]

coincides with \( x_{opt} \). Now choose \( \theta^0 = (1, 1, 1), j_{max} = 30, \epsilon = 10^{-16}, \epsilon_1 = 10^{-4} \) and \( \mu_{max} = 10 \) in order to apply the algorithms MPAA and IMPAA for various values \( \mu \) and \( \mu_0 \), respectively.

All the optimization problems (14) were solved on a ROBOTRON EC 1056 computer using the globally convergent derivative-free Gauss-Newton methods imple-
mented by the program DNLQ\textsuperscript{1}) in form of the FORTRAN program REGNLG\textsuperscript{2}). The required eigenvalues were computed by EISPACK routines (see [4]). A characteristic measure for the amount of computational work is the number of eigenvalue routine calls.

Table 1 provides the results of MPAA. The given number of iteration steps $j$ and of EISPACK 'calls is required for computing $\lambda$ with an $\epsilon_1$ precision. The achieved approximation of $x_{opt}$ and of the associated eigenvalues is in general still somewhat better.

Table 1: Results of MPAA

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>0.1</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>30</td>
<td>24</td>
<td>10</td>
<td>13</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Calls</td>
<td>fail</td>
<td>356</td>
<td>184</td>
<td>194</td>
<td>fail</td>
<td>fail</td>
</tr>
</tbody>
</table>

For $\mu = 0.5$ and $\mu = 2.0$ more than $j_{\text{max}} = 30$ iteration steps are required to complete the iteration. On the other hand, for $\mu \geq 3$ the iteration completely fails due to an extreme oscillation of the iterates.

Table 2 shows the cost reduction of the improved version IMPAA in dependence on $\mu_0$.

Table 2: Results of IMPAA

<table>
<thead>
<tr>
<th>$\mu_0$</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
<th>5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>Calls</td>
<td>120</td>
<td>113</td>
<td>109</td>
<td>137</td>
<td>157</td>
<td>fail</td>
</tr>
</tbody>
</table>

In order to avoid overestimating of $\mu_i$ during the starting phase of the iteration process, the initial guess $\mu_0 = 1$ may be recommended as a safe version. However, as Table 2 shows, IMPAA is not sensitive with respect to greater values $\mu_0$ if these values are not too large (cf. $\mu_0 = 5$). Also, if IMPAA begins with a non-monotonic strating phase ($\mu_0 = 2$, $\mu_0 = 3$), the results are fairly satisfactory.

Finally, in Table 3 we give an survey of the behaviour of $\mu_i$ in IMPAA. Thus the monotonicity behaviour of the iteration can be studied in detail.

Table 3: The development of iteration exponents in IMPAA

<table>
<thead>
<tr>
<th>$\mu_0 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
</tr>
<tr>
<td>$\mu_1$</td>
</tr>
<tr>
<td>$\mu_2$</td>
</tr>
<tr>
<td>$\mu_3$</td>
</tr>
</tbody>
</table>

\textsuperscript{1}) Program package "Nichtlineare Gleichungen" Techn. Univ. Dresden (GDR) 1977.

Note that a constraint (here, \( i = 3 \)): which is not active (\( \lambda_3 = 0 \)) leads to growing exponents \( \mu_1 \) throughout the iteration. This is really a good way to handle such multiplier components. The maximum value \( \mu_{\text{max}} \), however, is responsible for avoiding overflow and underflow effects during the iteration.

REFERENCES


Manuskripteingang: 24.11.1987

VERFASSER:

Prof. Dr. Bernd Hofmann
Technische Hochschule Zittau
Sektion Mathematik und Naturwissenschaften
PSF 261
DDR-8800 Zittau