Self-Validating Methods — Some Recent Developements

Christian Jansson and Siegfried M. Rump
Technische Universität
Hamburg-Harburg

Dedicated to Professor Dr. H. Brakhage on the occasion of his retirement

1 Introduction

In 1960, Brakhage [4] proposed a method for linear integral equations of the second kind which was truly pioneering and guiding in numerical analysis. He was the first who introduced multi-grid methods by describing a two-grid iteration which contains all basic principles of modern multi-grid schemes. Today, multi-grid methods are known to be the most efficient algorithms in a broad area of applications.

A second chief ingredient of this paper are the rigorous a posteriori error estimations. These error estimations permit to give an automatic proof for the existence and uniqueness of the solution for linear integral equations of the second kind, and to calculate very sharp and guaranteed bounds for the solution. Now, for algorithms which prove existence and uniqueness of solutions together with calculating guaranteed bounds, the term *self-validating method, inclusion method, or verification method* is customary. We emphasize that these important ideas were published six years before the appearence of Moore's book [22] *Interval Analysis* in 1966. Most people say that this book is the start of self-validating methods.

This paper presents a survey of some recent developments related to self-validating methods for large sparse nonlinear systems and global optimization problems. In Section 2 some basic results and notations of interval arithmetic are given. Especially, it is shown how guaranteed bounds for the range of functions may be calculated with so-called inclusion functions. In Section 3 rigorous error bounds for the solution of large sparse nonlinear systems are discussed, and in Section 4, a branch and bound algorithm for global optimization problems is described which uses inclusion functions, and involves local optimization algorithms.

2 Rigorous bounds for the range of a function

The notation $a \pm \Delta a$ of numbers afflicted with an error term goes (at least) back to C.F. Gauß [31]. The interval $A := a \pm \Delta a = [a - \Delta a, a + \Delta a]$ represents some numbers $\widetilde{a} \in A$. For such "inprecise numbers" the four basic operations are defined in the well-known way. For example, for $B := b \pm \Delta b$

$$A \cdot B = a \cdot b \pm (|\, a \,| \cdot \Delta b + \Delta a \cdot |\, b \,|). \tag{2.1}$$

This definition omits higher order terms. It leads to the fundamental result:

> For multiplication and division the *relative error* of the result equals the sum of the relative errors of the operands, for addition and subtraction the *absolute error* of the result equals the sum of the *absolute errors* of the operands.

Therefore, only addition or subtraction are subject to larger (relative) errors in the result, and this happens when the magnitude of the result is much less than the magnitude of the operands (cancellation).

In the following, we are interested in rigorous estimations of the range of a function including higher order terms. For this purpose, it is advantageous to use a left/right bound representation of intervals rather than midpoint/radius like in (2.1). This is because the midpoint of a product is not necessarily equal to the product of the midpoints of two intervals. The midpoint a of an interval $A = [a - \Delta a, a + \Delta a]$ is denoted by $\mathrm{mid}(A)$, and its radius Δa also by $\mathrm{rad}(A)$.

The most simple definition of the four basic interval operations $\circ \in \{+, -, \cdot, /\}$ between two intervals $A = [\underline{a}, \overline{a}]$, $B = [\underline{b}, \overline{b}]$ is

$$A \circ B := [\min\{\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b}\}, \max\{\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b}\}]. \tag{2.2}$$

It is clear that with few case distinctions a faster computational implementation is possible. The basic and most important property of interval operations is the *isotonicity.* That is

$$\forall\, a \in A \quad \forall\, b \in B: \qquad a \circ b \;\in\; A \circ B \qquad \text{for} \quad \circ \in \{+, -, \cdot, /\}. \tag{2.3}$$

This property is transitive and offers a method for estimating the range of a function. Suppose a function $f : \mathbb{R}^n \to \mathbb{R}^n$ can be expressed as an arithmetical expression consisting of the four basic operations $+, -, \cdot, /$. Then replacing every arithmetical operation by their corresponding interval operations (call this function F) preserves isotonicity and therefore

$$\forall x \in \mathbb{R}^n : \quad f(x) \in F(x), \tag{2.4}$$

which gives an estimation of the range of f over $X \subseteq \mathbb{R}^n$. F is called an *inclusion function* or *interval evaluation*. Define the set of interval vectors by $\mathbb{IR}^n = \mathbb{IR} \times \ldots \times \mathbb{IR}$. Then operations between real vectors $x, y \in \mathbb{R}^n$ can be lifted into $\mathbb{IR}^n$ preserving the fundamental property of interval analysis, the isotonicity (2.3). The same is true for interval matrices as well as for the corresponding extension over the field of complex numbers. For a detailed treatise of the subject see, among others, [2], [24].

The main point is that this process is rigorously implementable and executable on digital computers. Suppose a finite set of floating point numbers $\mathbb{F} \subseteq \mathbb{R}$ is given. Then with $\underline{a}, \overline{a} \in \mathbb{F}$, $\underline{a} \leq \overline{a}$, the floating point interval

$$[\underline{a}, \overline{a}] := \{ x \in \mathbb{R} \mid \underline{a} \leq x \leq \overline{a} \}$$

represents the set of all *real numbers* between $\underline{a}$ and $\overline{a}$. The IEEE 754 floating point standard [12] provides floating point operations with rounding downwards and rounding upwards. Using these in the definition (2.2) maintains the isotonicity (2.3), and therefore yields also the possibility of estimating the range of a function by (2.4) and (2.5). For more details see standard text books on interval analysis, among them [2], [24].

The observations made so far can be extended to transcendental functions as well. Especially, this is simple for monotone functions like the exponential function:

$$e^A := [e^{\underline{a}}, e^{\overline{a}}] \qquad \text{for all } A \in \mathbb{IR}.$$

For intervals with floating point end points, proper rounding has to be used. For other functions like sin, the extreme values are well known, and with the help of some case distinctions it is possible to implement interval extensions for exp, log, trig, inverse trig, hyperbolic, inverse hyperbolic, square root and more. This is even possible for complex intervals. The difficulty in this case is that extreme values are not necessarily in a vertex of the input interval. Descriptions for interval evaluations of real and complex trancendental functions can be found in [5], [17].

Summarizing, the approach described above computes estimations for the range of a function. However, due to data dependencies this is, in general, an overestimation of the true range.

Consider $f(x) = x^2 - 4x$ in the interval $X = [0,3]$. Then the approach above yields

$$f(X) \subseteq F(X) = X^2 - 4X \subseteq [0,3]^2 - 4 \cdot [0,3] \subseteq [0,9] - [0,12] \subseteq [-12,9].$$

This is a rigorous bound for the true range $f(X) = [-4,0]$ but, however, a huge overestimation. This is due to unresolved data dependencies, because the interval X occurs more than once in $F(X) = X^2 - 4X$. The aim of the paper is to derive techniques how to avoid or diminish such overestimations, or how to squeeze useful information out of them. So far we can state the following theorem.

Theorem 2.1. Let a function $f : \mathbb{R}^n \to \mathbb{R}^n$ be given by means of an expression consisting of constants, basic arithmetic operations, and standard functions like square root, exp, log trig, inverse trig, hyperbolic and inverse hyperbolic functions. Replacing every operation by its corresponding interval operation defines an inclusion function $F : \mathbb{IIR}^n \to \mathbb{IIR}^n$, the natural interval evaluation of f, with the following property. If for an interval vector $X \in \mathbb{IIR}^n$ the expression for F is executable, then

$$\forall\, x \in X : \quad f(x) \in F(X). \tag{2.5}$$

The assumption that F is executable in the assertion of Theorem 2.1 is necessary because, due to overestimation, it may occur that, for example, a denominator interval contains zero.

We want to give two examples for estimating the range of a function. The first, a one-dimensional example is a straightforward application of Theorem 2.1, the second example estimates the range of an implicitly given function. The latter will be used in chapter 4. In [1] the following formula for the logarithmic Gamma function is given:

$$\ln\Gamma(x) = (x - \tfrac{1}{2}) \cdot \ln x - x + \tfrac{1}{2}\ln 2\pi + \tfrac{1}{12x} - \tfrac{1}{360x^3} + \tfrac{\Theta}{1260x^5} \qquad \text{for} \quad 0 < x \in \mathbb{R},\ 0 < \Theta < 1. \tag{2.6}$$

Suppose we wish to compute the range of $\Gamma(x)$ over $X = [-0.505, -0.503]$, an interval containing an extreme value of Γ. Interval anaysis allows us to replace x by X and Θ by $[0,1]$. Combining this with $\Gamma(x) = \Gamma(x+1)/x$, which means $\Gamma(X) \subseteq \Gamma(X+1)/X$, yields an estimation for the range of Γ over X by simply replacing all operations and functions by the corresponding interval operations and functions. This process can be fully automized and needs no further knowledge on Γ like a Lipschitz constant or others. We obtain

$$\Gamma(X) = \{\Gamma(x) \mid x \in [-0.505, -0.503]\} \subseteq [-3.566, -3.523].$$

The true value for the range is $[-3.54467, -3.54464]$. The overestimation is due to data dependencies in the interval version of (2.6).

The second example is the computation of the smallest singular value of a symmetric matrix, where the data of the matrix are afflicted with tolerances. That is, for $[A] \in \mathbb{IR}^{n \times n}$ we look for

$$\varphi([A]) := \left\{ \min|\lambda| \;\middle|\; \exists\, A \in [A] \text{ with } A = A^T \text{ and } \lambda \text{ is eigenvalue of } A \right\}. \quad (2.7)$$

If $\widetilde{\lambda}, \widetilde{x}$ are approximations for an eigenvalue/eigenvector pair of some symmetric matrix A, then perturbation theory of eigenvalues of symmetric matrices [8] yiels the existence of an eigenvalue $\widehat{\lambda}$ of A with

$$|\widetilde{\lambda} - \widehat{\lambda}| \leq ||A\widetilde{x} - \widetilde{\lambda}\widetilde{x}||_2. \quad (2.8)$$

Furthermore, for symmetric E and $\lambda_i(A)$ being the i-th eigenvalue of A it is

$$|\lambda_i(A + E) - \lambda_i(A)| \leq ||E||_2. \quad (2.9)$$

Let $\widetilde{\lambda}_i, \widetilde{x}_i$ be approximations for eigenvalue / eigenvector pairs of $\text{mid}([A])$. For $E := \text{rad}([A])$ holds $A \in [A] \Leftrightarrow |A - \text{mid}([A])| \leq E$. Hence, for any symmetric $A \in [A]$ we have $||A - \text{mid}([A])||_2 \leq ||E||_2$ by Perron-Frobenius Theory and $E \geq 0$. If the intervals $\widetilde{\lambda}_i \pm ||A\widetilde{x}_i - \widetilde{\lambda}_i\widetilde{x}_i||$ are disjoint, then by (2.8) each of them contains exactly one eigenvalue of A. Then (2.9) implies that the union of

$$\widetilde{\lambda}_i \pm \Delta_i \quad \text{with} \quad \Delta_i = ||A\widetilde{x}_i - \widetilde{\lambda}_i\widetilde{x}_i||_2 + ||\text{rad}([A])||_2$$

contains the eigenvalues of all $A \in [A]$. Computing the minimum modulus yields bounds for $\varphi([A])$. The method yields very sharp estimations for $\varphi([A])$ because no data dependencies occur. It works, if all eigenvalues are distinct. If this is not the case, and X is a matrix of approximate eigenvectors, then the set of eigenvalues of $[B] := X^{-1} \cdot [A] \cdot X$ contains the eigenvalues of all $A \in [A]$ and bounds can be calculated by Gershgorin's Theorem. This is Lohners's method [19]. In fact, he replaces X^{-1} by X^T and estimates the error $X^{-1} - X^T$.

In the next chapter we will need a sharp lower estimate for $\varphi([A])$ for large sparse matrices. Such a method will be derived without using an approximation of some eigenvector.

The techniques used for the latter example reveals a basic principle of the inclusion methods we are looking for. Most of the calculations are performed in floating point, major parts of the analysis are performed a priori on a mathematical basis, and very few parts are actually performed in interval calculations. This is very much in the spirit of Wilkinson, the father of error analysis. He writes in 1971 [33]:

"In general it is the best in algebraic computations to leave the use of interval arithmetic as late as possible so that it effectively becomes an a posteriori weapon."

In general, numerical algorithms compute good approximations to the solution of a given problem. We close this chapter by giving a simple numerical example where standard numerical algorithm produce poor approximations. Consider

$$A = \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ & 1 & 1 & \\ & & \ddots & \ddots \end{pmatrix}, \quad B = \begin{pmatrix} 1 & \cdots & & 1 \\ & 1 & \cdots & 1 \\ & & \ddots & \vdots \\ & & & 1 \end{pmatrix},$$
$$nB^{-1} = \begin{pmatrix} 1 & & & -1 \\ & 1 & & -1 \\ & & \ddots & \vdots \\ & & & 1 \end{pmatrix}. \tag{2.10}$$

Then A as well as $B^{-1} \cdot A \cdot B$ has an n-fold eigenvalue $\lambda = 1$. The matrix $B^{-1}AB$ has integer entries not greater than 2 in modulus. Then EISPACK algorithms implemented in MATLAB [20] produce for $n = 17$ the following approximative eigenvalues plotted in the complex plane, without giving any warning.

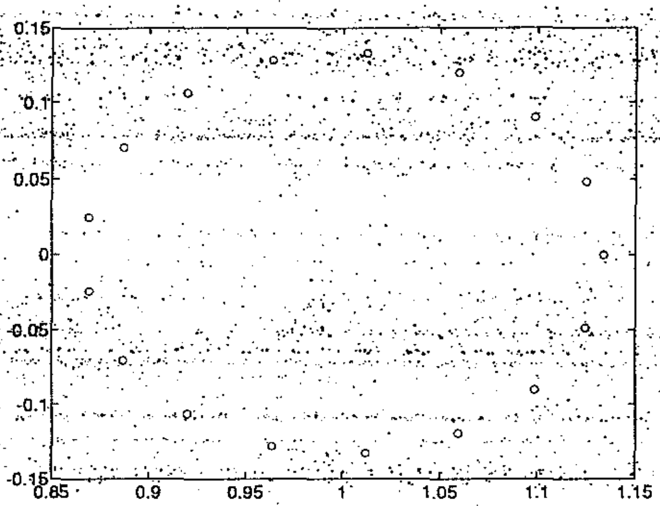


Figure 2.1. EISPACK approximations for eigenvalues of $B^{-1}AB$, $n = 17$

3 Bounds for the solution of large sparse nonlinear systems

In this chapter we will use the possibility to compute rigorous estimations of the range of a function to compute bounds for the solution of a nonlinear system. Let $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$, be continuously differentiable. For some $\widetilde{x} \in D$, which is supposed to be an approximate solution of $f(x) = 0$, let $R \in \mathbb{R}^{n \times n}$ be given such that $R \cdot f(\widetilde{x} + x) \approx 0$ for small x. Define $g : D' \subseteq \mathbb{R}^n \to \mathbb{R}$ with $D' = \{ x - \widetilde{x} \mid x \in D \}$ and

$$g(x) = x - R \cdot f(\widetilde{x} + x). \tag{3.1}$$

If for compact and convex $\emptyset \neq X \subseteq D'$

$$g(X) \subseteq X, \tag{3.2}$$

then Brouwer's Fixed Point Theorem implies the existence of some $\widehat{x} \in D'$ with

$$g(\widehat{x}) = \widehat{x} = \widehat{x} - R \cdot f(\widetilde{x} + \widehat{x}) \quad \text{and therefore } R \cdot f(\widetilde{x} + \widehat{x}) = 0.$$

If R can be shown to be nonsingular, then $f(\widetilde{x} + \widehat{x}) = 0$. Note that we did not need any assumptions on the quality of $\widetilde{x}$ or R. Condition (3.2) is suitable for the application of interval analysis if the range of g can be estimated without too much overestimation. Replacing every operation by its corresponding interval operation in (3.2) for $X \in \mathbb{IR}^n$, $X \subseteq D'$ yields an interval evaluation $F : \mathbb{IR}^n \to \mathbb{IR}^n$ with $x \in X \Rightarrow f(\widetilde{x} + x) \in F(\widetilde{x} + X)$. Because,

$$\operatorname{diam}(X - R \cdot F(\widetilde{x} + X)) = \operatorname{diam}(X) + \operatorname{diam}(R \cdot F(\widetilde{x} + X)),$$

except in trivial cases, (3.2) cannot be verified using this approach. Therefore, we use an expansion of f w.r.t. $\widetilde{x}$, that is, we locally linearize. Suppose, for given $\widetilde{x} \in D$, $X \in \mathbb{IR}^n$, $X \subseteq D'$ there is some $S(\widetilde{x}, X) \in \mathbb{IR}^{n \times n}$ with

$$\forall\, x \in X : \quad f(\widetilde{x} + x) \in f(\widetilde{x}) + S(\widetilde{x}, X) \cdot x. \tag{3.3}$$

Let $J(x) : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ be the Jacobian of f. Then it can be shown (cf. [22], [24]) that (3.3) is satisfied for $S(\widetilde{x}, X) := J(\widetilde{x} + X)$ provided X contains 0. Then

$$\begin{aligned} g(x) &\in x - R \cdot f(\widetilde{x}) - R \cdot S(\widetilde{x}, X) \cdot x = \\ &-R \cdot f(\widetilde{x}) + \{I - R \cdot S(\widetilde{x}, X)\} \cdot x. \end{aligned} \tag{3.4}$$

That means

$$-R \cdot f(\widetilde{x}) + \{I - R \cdot S(\widetilde{x}, X)\} \cdot X \subseteq X \quad \Rightarrow \quad g(X) \subseteq X. \tag{3.5}$$

Therefore, Brouwer's Fixed Point Theorem can be applied, and $\widetilde{x} + X$ is verified to contain a zero of f, if four problems are solved:

1) the computation of some $S(\widetilde{x}, X) \in \mathbb{IR}^{n \times n}$ with (3.3);

2) the computation of a suitable $R \in \mathbb{R}^{n \times n}$;

3) the verification of the regularity of R;

4) the verification of $-R \cdot f(\widetilde{x}) + \{I - R \cdot S(\widetilde{x}, X)\} \cdot X \subseteq X$.

The first problem can be solved by means of automatic differentiation [9], where the argument x is replaced by the interval vector $\widetilde{x} + X$, and all operations are replaced by their corresponding interval operations.

We want to stress that this process can be fully automized by means of automatic differentiation in conjunction with an operator concept [27]. The ansatz (3.3) goes back to the Krawczyk operator [18], see also [23]. We should mention that (3.3) is also possible for continuous but nondifferentiable functions. In this case, $S(\widetilde{x}, X)$ is replaced by a slope matrix. For details, see [29].

Next we will solve problems 2 to 4 for the n-dimensional case. We want to concentrate on large and sparse nonlinear systems. A possible choice of R is an approximate inverse of the midpoint of $S(\widetilde{x}, X)$. However, for large and sparse systems of nonlinear equations an approximate inverse would, in general, become full. This would result in a huge amount of memory and unacceptable computing times. We overcome this problem by the following theorem.

Nevertheless, we use the precise inverse of $\operatorname{mid}(S(\widetilde{x}, X))$ as the preconditioner. We do this by computing a lower bound of the smallest singular value of $\operatorname{mid}(S(\widetilde{x}, X))$, and thereby verifying a posteriori the regularity of $\operatorname{mid}(S(\widetilde{x}, X))$.

Theorem 3.1. Let continuous $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be given, $\widetilde{x} \in D$ and $0 < \rho \in \mathbb{R}$ with

$$X := \{x \in \mathbb{R}^n \mid ||x|| \leq \rho\} \quad \text{and} \quad \widetilde{x} + X \subseteq D. \tag{3.6}$$

Furthermore, let $S(\widetilde{x}, X) \in \mathbb{IR}^{n \times n}$ be given, and assume that

$$\forall\, x \in X : \quad f(\widetilde{x} + x) \in f(\widetilde{x}) + S(\widetilde{x}, X) \cdot x. \tag{3.7}$$

If $0 < \tau \in \mathbb{R}$ is a lower estimate on the smallest singular value $\sigma_{\min}$ of $\operatorname{mid}(S(\widetilde{x}, X))$ and

$$||f(\widetilde{x})|| + ||\operatorname{rad}(S(\widetilde{x}, X))|| \cdot \rho \leq \tau \cdot \rho, \tag{3.8}$$

then there exists some $\widehat{x} \in X$ with $f(\widetilde{x} + \widehat{x}) = 0$.

Remark. $||\cdot||$ denotes the 2-norm.

Proof. The matrix $R := \operatorname{mid}(S(\widetilde{x}, X))^{-1}$ is well-defined because $\sigma_{\min}\Big(\operatorname{mid}(S(\widetilde{x}, X))\Big) \geq \tau > 0$. Thus (3.7) yields

$$\forall\, x \in X : \quad x - R \cdot f(\widetilde{x} + x) \in -R \cdot f(\widetilde{x}) + \{I - R \cdot S(\widetilde{x}, X)\} \cdot x. \tag{3.9}$$

Let $M \in S(\widetilde{x}, X)$ be fixed but arbitrary. Then $|M - R^{-1}| \leq \operatorname{rad}(S(\widetilde{x}, X))$, and

$$-R \cdot f(\widetilde{x}) + \{I - R \cdot M\} \cdot x = R \cdot [-f(\widetilde{x}) + \{R^{-1} - M\} \cdot x].$$

Introducing norms and observing $||R|| \leq \tau^{-1}$ yields

$$||-R \cdot f(\widetilde{x}) + \{I - R \cdot M\} \cdot x|| \leq \tau^{-1} \cdot [||f(\widetilde{x})|| + ||\operatorname{rad}(S(\widetilde{x}, X))|| \cdot ||x||].$$

This is true for every $M \in S(\widetilde{x}, X)$ and every $x \in X$. Therefore (3.8) and (3.9) imply

$$\forall x \in X : \quad x - R \cdot f(\widetilde{x} + x) \in X.$$

Hence, Brouwer's Fixed Point Theorem implies the existence of a fixed point $\widehat{x} \in X$ with $\widehat{x} - R \cdot f(\widetilde{x} + \widehat{x}) = \widehat{x}$. The regularity of R proves $f(\widetilde{x} + \widehat{x}) = 0$ and the theorem. ∎

Our verification process is two-fold: first, the lower bound τ has to be computed and verified and second, condition (3.8) has to be verified for a suitable X.

A verified lower bound on the smallest singular value of a matrix which implies a verified condition estimator can be calculated according to [30]. Again, very much in the spirit of Wilkinson, almost only floating point arithmetic is used in conjunction with a priori and a posteriori error estimates.

Let $PQR \approx A$ be an (approximate) floating point decomposition of a matrix A, for example an LU- or LDL^T-decomposition. Then

$$\sigma_{\min}(PQR) \geq \sigma_{\min}(P) \cdot \sigma_{\min}(Q) \cdot \sigma_{\min}(R). \tag{3.10}$$

Together with an upper bound for $||A - PQR||$ it suffices to derive a method for estimating the smallest singular value of a nonsingular triangular matrix L (see [30]).

Denote the eigenvalues of a symmetric matrix A by $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$ and its singular values by $\sigma_1(A) \geq \cdots \geq \sigma_n(A)$. If L is nonsingular, then the matrix LL^T is symmetric positive definite (s.p.d.) and $\lambda_i(LL^T) = \sigma_i(L)^2$ for $1 \leq i \leq n$. Then $LL^T - \lambda I$, $0 \leq \lambda \in \mathbb{R}$ is s.p.d. if and only if $\sigma_n(L)^2 = \lambda_n(LL^T) > \lambda$. But $LL^T - \lambda I$ is s.p.d. iff a (nonsingular) Cholesky factor G with $GG^T = LL^T - \lambda I$ exists. An approximation for $\lambda_n(LL^T)$ can be obtained by inverse power iteration and forward and backward substitution with L and L^T.

To prove that $LL^T - \lambda I$ is s.p.d. might be performed by an interval version of Cholesky's decomposition. If it succeeds, i.e. all diagonal elements stay positive, then $LL^T - \lambda I$ is proved to be s.p.d. However, due to overestimation and data dependencies, this process may not work even for small and simple examples. Consider $A = LL^T - \lambda I$ with

$$L = 0.9 \cdot \begin{pmatrix} 1 & & & & \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ & 1 & 1 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots \end{pmatrix} \quad \text{and} \quad \lambda = 0.$$

The factor 0.9 assures that the elements of the Cholesky factor L are not exactly representable on a digital computer. Then interval Cholesky decomposition (every operation is replaced by its corresponding interval operation) for dimension n produces a factor $[G] \in \mathbb{IR}^{n\times n}$ with a final diagonal element $[G]_{nn}$ of the following diameter.

n	10	15	20	25	30	35	40
diam$([G]_{nn})$	$6\cdot 10^{-12}$	$7\cdot 10^{-10}$	$8\cdot 10^{-8}$	$8\cdot 10^{-6}$	$2\cdot 10^{-3}$	$2\cdot 10^{-1}$	failed
cond(LL^T)	$1.5\cdot 10^{2}$	$3.3\cdot 10^{2}$	$5.9\cdot 10^{2}$	$8.5\cdot 10^{2}$	$1.2\cdot 10^{3}$	$1.7\cdot 10^{3}$	$2.1\cdot 10^{3}$

The reason for the failure of this approach is overestimation and data dependencies, not the condition of the problem.

Again, very much in the spirit of Wilkinson, we calculate an approximate Cholesky decomposition $\widetilde{G}\widetilde{G}^T$ of $LL^T - \lambda I$ and estimate the error. Then perturbation theory [8] yields

$$|\lambda_i(\widetilde{G}\widetilde{G}^T) - \lambda_i(LL^T - \lambda I)| \le \|\widetilde{G}\widetilde{G}^T - (LL^T - \lambda I)\| =: \nu$$

and

$$\sigma_n(L)^2 = \lambda_n(LL^T) \ \ge\ \lambda - \nu.$$

This proves the following theorem [30].

Theorem 3.2. Let $L \in \mathbb{R}^{n\times n}$, $\lambda \in \mathbb{R}$ and $\widetilde{G} \in \mathbb{R}^{n\times n}$ be given and define

$$\nu := \|\widetilde{G}\widetilde{G}^T - (LL^T - \lambda I)\|.$$

If $\lambda \ge \nu$, then (3.11)

$$\sigma_n(L) \ge (\lambda - \nu)^{1/2}. \tag{3.11}$$

In Theorem 3.1, we choose the preconditioning matrix to be the inverse of $\mathrm{mid}(S(\widetilde{x}, X))$ itself. The nonsingularity of this matrix is shown a posteriori using (3.10) and (3.11). In the final inclusion formula (3.8) only the radius of $S(\widetilde{x}, X)$ occurs, not $S(\widetilde{x}, X)$ itself. This yields good bounds in a norm-wise sense.

We illustrate the application of Theorem 3.2 by a simple example. Consider Emden's equation [26]

$$-\Delta U = U^2 \quad \text{on} \quad \Omega := (0, l^{-1}) \times (0, l), \quad U = 0 \quad \text{on} \quad \partial\Omega. \tag{3.12}$$

Using central differences to approximate ΔU and m, n inner points along the axes, resp., we obtain a discretized problem

$$A \cdot u + u^2 = 0 \quad \text{with} \quad A \in \mathbb{R}^{mn \times mn} \tag{3.13}$$

in $m \cdot n$ variables which has a sparse structure. We use

$$\widetilde{f}(x_1, x_2) := \lambda \cdot x_1^2(1 - x_1)^2 \cdot x_2^2(1 - x_2)^2$$

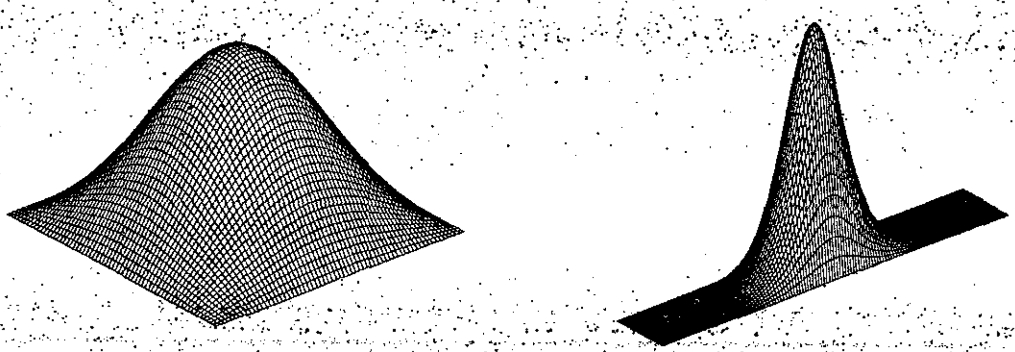
as an approximate solution of (3.13) with a suitable value for λ as a starting value for a simplified Newton iteration. Let $\widetilde{u}$ denote an approximate solution of the discretized problem (3.13) obtained in such way, and let $\widehat{u}$ denote a true solution. Then we verify

$$\|\widehat{u} - \widetilde{u}\| \leq e \cdot \|\widetilde{u}\| \tag{3.14}$$

where e is given in the following table. All computations are performed in double precision equivalent to 17 decimal places. Furthermore, we use the abbreviations

N	dimension of the discritized nonlinear system (3.13);
m	inner grid points in first direction;
n	inner grid points in second direction (that is the bandwidth);
iter	number of inverse power iterations to approximate σ_{min};
u_m	$\widehat{u}(\frac{1}{2}l^{-1}, \frac{1}{2}l)$;
cond	condition number of $\text{mid}(S(X))$;
e	maximum relative error of inclusion according to (3.14).

The following pictures show solutions for $l = 1$ and $l = 2.5$.



Graph 3.1. Solution of Emden's equation (3.12) for $l = 1$ and $l = 2.5$

l	N	m	n	iter	u_m	cond	e
1.0	3969	63	63	3	29.3	$1.3 \cdot 10^3$	$6.2 \cdot 10^{-14}$
	16129	127	127	3	29.3	$5.2 \cdot 10^3$	$2.4 \cdot 10^{-13}$
	8001	127	63	3	29.3	$3.3 \cdot 10^3$	$1.9 \cdot 10^{-13}$
	32385	255	127	3	29.3	$1.3 \cdot 10^4$	$7.7 \cdot 10^{-13}$
2.0	3969	63	63	2	71.6	$1.7 \cdot 10^6$	$9.6 \cdot 10^{-11}$
	16129	127	127	2	71.5	$6.8 \cdot 10^6$	$3.8 \cdot 10^{-10}$
	8001	127	63	2	71.5	$2.0 \cdot 10^6$	$1.1 \cdot 10^{-10}$
	32385	255	127	2	71.5	$8.0 \cdot 10^6$	$4.5 \cdot 10^{-10}$
2.5	3969	63	63	2	111.9	$1.8 \cdot 10^9$	$9.8 \cdot 10^{-8}$
	16129	127	127	2	111.8	$7.3 \cdot 10^9$	$4.4 \cdot 10^{-7}$
	8001	127	63	2	111.8	$2.0 \cdot 10^9$	$1.0 \cdot 10^{-7}$
	32385	255	127	2	111.7	$7.9 \cdot 10^9$	$4.6 \cdot 10^{-7}$

Table 3.1. Verified inclusion for discretized Emden equation

The computational effort for the entire verification process is approximately equal to the time for the floating point decomposition of A. For larger values of l the condition numbers of the nonlinear system goes beyond the critical value of about 10^{10}. Above this value, higher computational precision is necessary. This can be seen as follows.

A good approximate solution $\widetilde{u}$ of (3.13) satisfies $||f(\widetilde{u})|| \approx \epsilon \cdot ||\widetilde{u}||$, where ϵ is the relative rounding error unit, in our case $\epsilon = 10^{-17}$. Let X be defined as in (3.6) and $S(\widetilde{x}, X)$ the Jacobian of (3.13).Then $\mathrm{rad}\big(S(\widetilde{x}, X)\big) = 2 \cdot h_x \cdot h_y \cdot \rho \cdot ||\widetilde{u}||$ and (3.8) writes

$$\epsilon \cdot ||\widetilde{u}|| + 2 \cdot h_x \cdot h_y \cdot \rho \cdot ||\widetilde{u}|| \cdot \rho \leq \tau \cdot \rho. \tag{3.15}$$

Table 3.1 gives $2 \cdot h_x \cdot h_y \cdot ||\widetilde{u}|| \approx 10^{-2}$, and for $||\widetilde{u}|| \approx 100$ and (3.14) writes

$$10^2 \cdot \epsilon/\rho + 10^{-2}\rho \leq \tau. \tag{3.16}$$

Solving (3.16) for positive ρ yields $\tau \lesssim 6 \cdot 10^{-9}$ or $cond(mid(S(\widetilde{x}, X))) \lesssim 10^{10}$. This is the inherent maximum condition number for Emden's equation which can be treated by our fixed point ansatz in double precision. Table 3.1 shows that inclusions up to this condition number are achieved.

For larger values of l, the problem becomes ill-conditioned, and sometimes this is difficult to detect. Denote successive iterates of some simplified or Quasi-Newton iteration by $\widetilde{u}^k$. Then a frequently used stopping criterion is $||f(\widetilde{u}^k)|| \leq c \cdot \varepsilon \cdot ||\widetilde{u}^k||$ for some small constant c. For the discretized Emden equation with $m = 127$, $n = 31$, which means N=3937 unknowns, we obtained for $l = 3.1$ the following results.

k	$\|\|f(\widetilde{u}^k)\|\| \;/\; \|\|\widetilde{u}^k\|\|$
1	$1.3 \cdot 10^{-3}$
2	$9.8 \cdot 10^{-5}$
3	$2.6 \cdot 10^{-7}$
4	$3.1 \cdot 10^{-12}$

Table 3.2. Simplified Newton-iteration for $l = 3.1$, $N = 3937$, $n = 31$.

These figures seem to indicate convergence behaviour and many stopping criterions would detect convergence for k=4. However, our verification process did not succeed, which looks strange at the first sight. Going one iteration further yields

k	$\|\|f(\widetilde{u}^k)\|\| \;/\; \|\|\widetilde{u}^k\|\|$
5	$1.2 \cdot 10^{-4}$

An inclusion method prevented us from accepting $\widetilde{u}^4$ as an approximation.

4 A branch and bound method for global optimization problems

In this section, we consider a branch and bound algorithm for the global optimization problem

$$Min\{f(x) \mid x \in X\}, \; X := \{x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x}\} \tag{4.1}$$

which is based on the tools of interval arithmetic. Here, $f : X \to \mathbb{R}$, the set of feasible points X is a box or interval vector with $\underline{x} \leq \overline{x}$, and $\leq$ is to be understood componentwise. We denote the global minimum (if it exists) by f^* and the set of global minimum points by X^*. Our algorithm calculates approximations of f^* and X^* and verified bounds for f^* and X^*; that is, these bounds are proved to be correct, including all rounding errors and higher order terms.

Branch and bound methods for solving global optimization problems (cf. for example [10], [11], [25], [28]) differ in the way they (i) partition the set of feasible solutions X into subregions, (ii) calculate bounds for the range of f on those subregions, and (iii) discard subregions for avoiding exhaustive search. We use in our method inclusion functions for bounding the range of f on subregions. For discarding subregions, we incorporate local optimization algorithms in a way that very early an approximation $\widetilde{x}$ of a global minimum point (or at least of a local minimum point) is calculated. The knowledge of such an approximation has important influence upon the efficiency: subregions Y of X contain no global minimum points and can be discarded if a lower bound of f on Y is greater than $f(\widetilde{x})$ for some $\widetilde{x} \in X$. Hence, the early knowledge of an approximation $\widetilde{x} \in X$ with $f(\widetilde{x}) \approx f^*$ yields to an efficient discarding technique.

Two main difficulties appear when local optimization algorithms are incorporated. First, local optimization algorithms require a starting point in an attraction region to converge to a global minimum point or at least to a stationary point of f. In many cases these attraction regions are small. Moreover, rounding errors and approximation errors (for example finite difference approximations) may produce wrong results. In order to demonstrate these facts and motivate the calculation of guaranteed bounds, we look at the following small application that is described in the book of Becker and Wittmer 1983 [3].

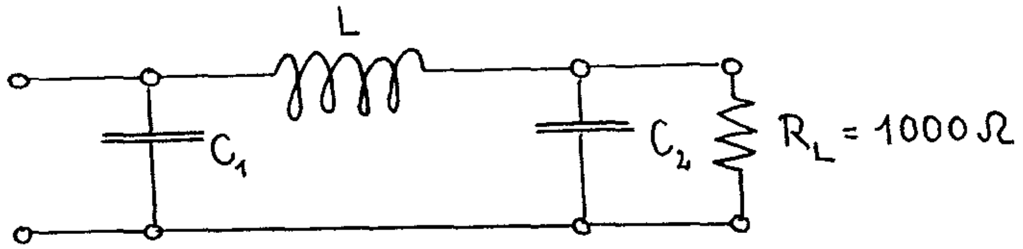


Figure 4.1

There, the problem is to minimize the costs of a voltage stabilizer (see Figure 4.1) such that the ripple factor $r = 3450/R_L C_1 C_2 L \leq 10^{-2}$. The costs of the components of the voltage stabilizer are determined from list prices, where the costs of the capacities

$$(0.25 + 2.5 \cdot 10^{-4})\frac{C}{\mu F}\ DM$$

are approximated by linear interpolation, and the costs for the inductivity

$$(1.0 + 5.0 \cdot 10^{-5} \frac{L^2}{H^2})\ DM$$

are approximated by quadratical interpolation (F: Farad, H: Henry, DM: German Mark). Summing up the costs by using the equations $3450/R_L C_1 C_2 L = 10^{-2}$, $R = 10^3 \Omega$, and dropping the constant term (this term has no influence on the optimal solution) yields the objective function

$$f(C_1, C_2) = 2.5 \cdot 10^{-4}(C_1 + C_2) + \frac{59.5}{C_1^2 C_2^2}$$

which has to be minimized subject to the technical constraints $C_1 > 0,\ C_2 > 0$. This is a strictly convex problem which has exactly one stationary point being the unique global minimum. Notice that the objective function is a simple rational function involving no standard functions. It is easy to verify the guaranteed bounds

$$13.639184346 \le C_1^*, C_2^* \le 13.639184374,$$

$$0.00852449021 \le f^* \le 0.00852449024$$

for the global minimum C_1^*, C_2^* (measured in μF) and the global minimum value f^* with an appropriate inclusion method.

For solving this problems we used the MATLAB routine *fminu* which is a BFGS-quasi-Newton method. All the runs (cf. Figure 4.2) were executed in double precision on a IBM RS/6000 by using MATLAB Version 4.1.1, Oct. 1993

starting point		calculated approximation of the optimal solution		
C_1	C_2	C_1^*	C_2^*	f^*
0.01	0.01	$4.6088 \cdot 10^8$	$4.6088 \cdot 10^8$	$2.3044 \cdot 10^5$
0.1	0.5	184.4824	37.3765	0.0555
10	10	13.6622	13.6622	0.0085
8	1	*	*	*
20	20	*	*	*

Figure 4.2

*: MATLAB gives the warning: "Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND=1.288638e-300."

Obviously, the routine *fminu* calculates approximations of the optimal solution which are completely wrong in the sense of forward and backward error analysis without giving any warning, and, for some starting points, *fminu* gives warnings that a well-conditioned problem is ill-conditioned. Thus, this BFGS-quasi-Newton method, one of the best algorithms for twice continuously differentiable functions, has problems for very simple applications with practical (not especially constructed) models and input data.

The BFGS-quasi-Newton method E04JAF in NAG's library is better, but in principle we have the same behaviour; for example the starting point $C_1 = C_2 = 25\mu F$ produces a "core dumped". Moreover, we see that the attraction region is very small. In our method, we use inclusion functions (interval arithmetic evaluations) to produce good starting points for local optimization algorithms. To do this, we incorporate subdivision scheme to be described in the following, and use the heuristic that in many cases subregions with the smaller lower bound contain smaller function values.

The second difficulty is when to call a local optimization algorithm in a global optimization method. Ideally, the local optimization algorithm should be called, if then an approximation of a global minimum point is calculated. We attempt to handle this problem by our branch and bound scheme and a special rule for calling a local optimization algorithm. In the following, we will describe an simplified version of our method in the one-dimensional case. For the more complicated multi-dimensional case, the reader is referred to [13]. The method consists of two algorithms: algorithm BRANCH AND BOUND manages the partitioning of the feasible domain X, and calls procedure SUBDIVISION which is responsable for selecting starting points, for calling the local optimization algorithm, for splitting the problem into subproblems, and for discarding subregions.

For the remainder of this section, we assume that an inclusion function F of f is given, and $\underline{F}^*, \overline{F}^*$ denote a lower and an upper bound of the global minimum value f^*, respectively. Moreover, let $n_{it}, n_d \in \mathbb{N} \setminus \{0\}$.

Algorithm BRANCH AND BOUND starts with the original problem which is stored on list S, and where the bounds of the global minimum value are set to $-\infty, \infty$ (cf. (4.2),(4.3)). On list A, which is empty at the beginning, approximations of the global minimum points and the global minimum value will be stored (cf. (4.21)). The algorithm continues by removing all subproblems from list S, and by applying procedure SUBDIVISION to each subproblem (cf. (4.4) to (4.8)). Then in (4.9) the lower bound $\underline{F}^*$ is updated. Obviously, this bound is equal to the minimum of all lower bounds of the subproblems which are not discarded and stored on our pool of subproblems list S. The algorithm terminates after n_{it} iteration steps, or if in (4.10) the lower and the upper bound are sufficiently close.

In procedure SUBDIVISION the given box Y is bisected (cf. (4.16)), and with the inclusion function F lower bounds of the range of f on Y are calculated. Therefore, we get two subproblems. Now, a main idea is to proceed with the subproblem which has the smaller lower bound (cf. (4.17)). This is motivated by the heuristic argument that in many cases a subproblem with a smaller lower bound also contains feasible points with smaller function values. This way of bisecting is continued until the width of box Y $(w(Y) := \overline{Y} - \underline{Y})$ is equal w^*, and in this case the local optimization algorithm is called provided that for the starting point mid(Y) the corresponding function value is smaller then $\overline{F}^*$ (cf. (4.20), (4.21)). Using the above heuristic argument, it follows that this way of bisecting yields improved starting points for the local optimization algorithm. Subproblems with greater lower bounds are stored on a working list W (cf. (4.19)) provided their width are greater w^*, and are stored on list L (and therefore given back to the pool of subproblems list S) provided that their width is equal to w^* (cf. (4.23)). Obviously, subproblems with a lower bound greater than $\overline{F}^*$ cannot contain a global minimum point, and thus are discarded in steps (4.14),(4.19),(4.23).

```
algorithm BRANCH AND BOUND $(X, S, A, \underline{F}^*, \overline{F}^*)$ begin
  $Y^{(1)} := X,\ \underline{F}^* := -\infty,\ \overline{F}^* := \infty,\ F(Y^{(1)}) := [-\infty, \infty]$;        (4.2)
  initialize lists $S := \{(Y^{(1)}, F(Y^{(1)}))\}, A := \emptyset$;                                         (4.3)
    for $i = 1, \cdots, n_{it}$ do begin                                                                      (4.4)
    $S' := S,\ S := \emptyset$;                                                                               (4.5)
    for all pairs $(Y, F(Y)) \in S$ do begin                                                                  (4.6)
      call SUBDIVISION $(Y, F(Y), L, \overline{F}^*, A)$;                                                    (4.7)
      append list $L$ at the end of list $S$                                                                  (4.8)
    end;
    $\underline{F}^* := \text{Max}\{\underline{F}^*, \text{Min}\{\underline{F}(Y) \mid (Y, F(Y)) \in S\}\}$;   (4.9)
    if $\overline{F}^* - \underline{F}^* < \epsilon$ then STOP                                               (4.10)
  end;
end;
```

```
procedure SUBDIVISION $(Y, F(Y), L, \overline{F}^*, A)$; begin
  initialize lists $W := \{(Y, F(Y))\}, L := \emptyset$, and set $w^* := w(Y)/2^{nd}$;          (4.11)
  while $W \neq \emptyset$ do begin                                                              (4.12)
    remove the last pair $(Y, F(Y))$ from list $W$;                                             (4.13)
    if $\underline{F}(Y) < \overline{F}^*$ then                                                 (4.14)
    while $w(Y) > w^*$ do begin                                                                  (4.15)
      bisect $Y$ such that $Y = Y^{(1)} \cup Y^{(2)}$ and calculate $F(Y^{(1)}), F(Y^{(2)})$    (4.16)
      (we assume $\underline{F}(Y^{(1)}) < \underline{F}(Y^2)$, otherwise exchange the indices);
      $Y := Y^{(1)}, F(Y) := F(Y^{(1)})$;                                                       (4.17)
      if $w(Y^{(1)}) > w^*$ then                                                                 (4.18)
        if $\underline{F}(Y^{(2)}) \leq \overline{F}^*$ then enter $(Y^{(2)}, F(Y^{(2)}))$ at the end of list $W$; (4.19)
      else begin
        if $L = \emptyset$ and $f(\mathrm{mid}(Y)) < \overline{F}^*$ then                       (4.20)
          call a local optimizition algorithm with                                               (4.21)
          starting point mid$(Y)$ that calculates approximations
          $\widetilde{x}, f(\widetilde{x})$, and append the pair $(\widetilde{x}, f(\widetilde{x}))$ at the
          end of list $A$;
        $\overline{F}^* := \mathrm{Min}\{\overline{F}(\widetilde{x}), \overline{F}^*\}$;        (4.22)
        $\underline{F}(Y^{(j)}) \leq \overline{F}^*$ then append $(Y^{(j)}, F(Y^{(j)}))$ at the end list $L$ $(j = 1, 2)$;
(4.23)
      end;
    end;
  end;
end;
```

Next, we state a theorem concerning some properties of the above method:

Theorem 4.1: Let $f: X \to \mathbb{R}$, and assume further that

(a) F is an inclusion function of f;

(b) $n_{it}, n_d \in \mathbb{N}\backslash\{0\}$;

(c) the local optimization algorithm terminates after a finite number of steps.

Then the following results hold:

(i) Algorithm BRANCH AND BOUND terminates after a finite numbers of steps;

(ii) $\underline{F}^* \leq f^* \leq \overline{F}^*$;

(iii) list A contains an approximation $\widetilde{x}$ such that $\underline{F}^* \leq f(\widetilde{x}) \leq \overline{F}^*$;

(iv) $X^* \subseteq U\{Y \mid (Y, F(Y)) \in S\}$

This result shows that guaranteed bounds for the global minimum value and the global minimum points are calculated, and moreover, (iii) shows that a calculated approximation $\widetilde{x}$ is contained in the level set $\{\, x \mid f(x) - f^* \leq \overline{F}^* - \underline{F}^* \,\}$. For a proof of Theorem 4.1 and for some convergence results which describe the behaviour for $i \to \infty$ see [13]. Especially, it can be shown that for increasing n_{it}, n_d the bounds converge to f^*, X^* provided that some mild assumptions are fulfilled.

So far, this branch and bound method uses no derivatives, and it is suitable for problems which are not differentiable, or where derivatives are not available. Especially, the bounds calculated for X^* may be very rough. Inclusion functions of the first and second derivatives can be used to accelerate this branch and bound scheme, and to calculate very sharp bounds for f^* and X^*. For details of how to incorporate derivatives into this branch and bound scheme, we refer the reader to [13]. In the following we give numerical examples of our method for some multi-dimensional problems. Numerical experiments of about 50 problems are described in [14] and [15]. These problems include the well-known test sets of Dixon and Szegö [7] and Hansen [10]. In the first report the results are calculated without using derivatives, whereas in the second report inclusion functions of the first and second derivatives are used.

The method has been implemented by using the C++ library PROFIL [16].The corresponding inclusion functions are natural interval evaluations.

We use the following abbreviations:

n_m	denotes the number of global minima;
n_L	denotes the number of calls of the local optimization algorithm;
l_s	denotes the maximal length of the List S, and indicates the storage requirement;
n_{rf}, n_{if}	are the total number of real and inclusion function calls, of f, respectively;
n_{ig}, n_{ih}	are the total number of inclusion gradient and hessian calls, respectively;
t_{STU}	is the total execution time. The unit for t_{STU} is the time needed to perform 1000 calls of the Shekel Function No. 5 at (4,4,4,4). On a SparcServer 330 one unit is 0.2s.

Example 1: Griewank's function [32]

$$f_G(x) = \sum_{i=1}^{n} x_i^2/d - \prod_{i=1}^{n} \cos\frac{x_i}{\sqrt{i}} + 1$$

$$-600 \le x_i \le 600, \quad d = 4000,$$

$$f^* = 0, \quad x^* = (0,\dots,0)$$

For dimension n=10 this function has several thousand local minima, and has the global minimum point $x^* = 0$ with global minimum value $x^* = 0$. In Törn and Zilinskas [32], page 195, test results of two methods are given:

Method	n_M	n_{rf}	STU
Griewank (1981)	–	6600	–
Snyman, Fatti (1987)	1	23399	90

Both methods give no guarantee, and Griewank's method has calculated only a local minimum.

Without using derivatives our method has calculated an approximation of the unique global minimum point $x^* = 0$ with magnitude of about 10^{-16}. The following table gives for some different n_{it}, n_d lower bound upper bounds for f^* and the computational costs.

n_{it}	n_d	$\underline{F}^*$	$\overline{F}^*$	n_M	n_L	l_s	n_{rf}	n_{if}	t_{STU}
2	8	0	$1.31006 \cdot 10^{-14}$	1	1	1	135	341	2.667
2	10	0	$1.80300 \cdot 10^{-13}$	1	1	1	417	421	4.600
2	12	0	$3.25184 \cdot 10^{-13}$	1	1	1	400	501	5.133

The bounds for x^* are very rough. By using natural inclusion functions of the first and second derivative of f, we get the following bounds for f^* and x^*:

$$f^* \quad \subseteq \quad [0, 4.551914400963142e \cdot 10^{-15}],$$

$$x^* \quad \subseteq \quad \begin{pmatrix} [\ -1.752875876867703 \cdot 10^{-15}, & 9.10327788226753 \cdot 10^{-16}\ ] \\ [\ -2.49108923735287 \cdot 10^{-15}, & 1.273368348184274 \cdot 10^{-15}\ ] \\ [\ -3.724949921605188 \cdot 10^{-15}, & 8.832483289507194 \cdot 10^{-16}\ ] \\ [\ -3.68400804229635 \cdot 10^{-15}, & 1.634425727320358 \cdot 10^{-15}\ ] \\ [\ -1.364848010712496 \cdot 10^{-23}, & 1.282129949457193 \cdot 10^{-23}\ ] \\ [\ -2.9778502051909 \cdot 10^{-23}, & 2.646977960169689 \cdot 10^{-23}\ ] \\ [\ -5.156808186946726 \cdot 10^{-15}, & 1.868301654538232 \cdot 10^{-15}\ ] \\ [\ -5.743642108837937 \cdot 10^{-15}, & 1.762776141781241 \cdot 10^{-15}\ ] \\ [\ -6.995533733778533 \cdot 10^{-15}, & 9.622619705035564 \cdot 10^{-16}\ ] \\ [\ -1.77843831698901 \cdot 10^{-23}, & 1.613002194478404 \cdot 10^{-23}\ ] \end{pmatrix}$$

with

n_{it}	n_d	n_M	n_L	l_S	n_{rf}	n_{if}	n_{ig}	n_{ih}	t_{STU}
≥ 2	10	1	1	1	416	261	1	7	9.25
≥ 2	15	1	1	1	163	321	1	7	8.25

In the first column $n_{it} \geq 2$ means that these results hold for all those n_{it}.

For dimension $n = 50$ Griewank's function has millions of local minima, and we get the following results without using derivatives

n_{it}	n_d	$\underline{F}^*$	$\overline{F}^*$	n_M	n_L	l_s	n_{rf}	n_{if}	t_{STU}
1	10	0	$1.14087 \cdot 10^{-12}$	1	1	1	7644	1101	110.333
2	10	0	$1.14087 \cdot 10^{-12}$	1	1	1	7645	2101	140.333
1	15	0	$2.25375 \cdot 10^{-14}$	1	1	1	743	1601	48.067

Using inclusion functions of the derivatives we get

$$f^* \quad \subseteq \quad [0, 3.164135620181696 \cdot 10^{-14}],$$

$$x^* \quad \subseteq \quad \begin{pmatrix} [-1.804243928668264 \cdot 10^{-15}, 8.589598111826399 \cdot 10^{-16}] \\ \vdots \\ [-1.513044566938841 \cdot 10^{-14}, 3.251126042831056 \cdot 10^{-15}] \end{pmatrix},$$

and

n_{it}	n_d	n_M	n_L	l_S	n_{rf}	n_{if}	n_{ig}	n_{ih}	t_{STU}
≥ 2	15	1	1	1	767	1602	1	7	643.05
≥ 2	20	1	1	1	516	2102	1	7	729.15

Example 2: This example is given in Moore, Hansen and Leclerc [21]

The authors write that in the field of chemistry, a very common problem is to reconstruct a curve that is given by a n points (x_i, y_i), $i = 1, \ldots, n$. Normally, the curve is a sum of peaks, and the chemist desires to resolve the shape and the position of the individual peaks. They considered a curve which was the sum of two Gaussian peaks with:

$$\begin{aligned}
x_i &= 4 + (i+1)/10, \; i = 1, 2, \ldots, 81 \, , \\
y_i &= a_1 \cdot e^{-(\frac{x_i - u_1}{s_1})^2} + a_2 \cdot e^{-(\frac{x_i - u_2}{s_2})^2} \, , \\
a_1 &= 130.89 \quad , \quad a_2 = 52.6 \, , \\
u_1 &= 6.73 \quad , \quad u_2 = 9.342 \, , \\
s_1 &= 1.2 \quad , \quad s_2 = 0.97 \, .
\end{aligned}$$

The aim is to recover the six parameters a_1, a_2, u_1, u_2, s_1, s_2 of the curve function

$$p(x, a_1, a_2, u_1, u_2, s_1, s_2) = \sum_{j=1}^{2} a_j \cdot e^{-(\frac{x - u_j}{s_j})^2}$$

by minimizing the function

$$f(a_1, a_2, u_1, u_2, s_1, s_2) = \sum_{i=1}^{81} \left(a_1 e^{-(\frac{x_i - u_1}{s_1})^2} + a_2 e^{-(\frac{x_i - u_2}{s_2})^2} - y_i \right)^2 ,$$

where range of the six parameters was defined as follows:

$$\begin{aligned}
&[a_1] = [130, \; 135], \quad [a_2] = [50, \; 55], \quad [u_1] = [6, \; 8], \quad [u_2] = [8, \; 10], \\
&[s_1] = [1, \; 2], \quad [s_2] = [0.5, \; 1]
\end{aligned}$$

The method described in [21] uses interval derivatives in an extensive way. They obtained the following guaranteed bounds for the global minimum point

$$\begin{aligned}
a_1 &= [130.889999624668920, \; 130.890000237423440] \\
a_2 &= [52.5999994426222910, \; 52.6000003353821410] \\
u_1 &= [6.72999999580056230, \; 6.73000000523584680] \\
u_2 &= [9.34199999170696670, \; 9.34200000792551850] \\
s_1 &= [1.19999999502502950, \; 1.20000000672384770] \\
s_2 &= [0.96999998507893725, \; 0.97000001469388031]
\end{aligned}$$

and for the global minimum value

$$[\underline{f}^*, \overline{f}^*] = [6.3015390640982946 \cdot 10^{-13}, \; 9.9696829305332294 \cdot 10^{-11}].$$

For the guaranteed bounds of f^* given in [21] supposedly there is a misprint and the lower bound possibly should be negative.

The number of guaranteed decimal digits for the global minimum point varies between 7 and 9 and the time needed on a SUN SparcStation is reported as 109240s.

Our method without using derivatives gives the following results:

n_{it}	n_d	$\underline{F}^*$	$\overline{F}^*$	n_M	n_L	l_s	n_{rf}	n_{if}	t_{STU}
1	1	0	$2.72132 \cdot 10^{-20}$	1	1	8	256	53	12.4
2	1	0	$2.72132 \cdot 10^{-20}$	1	1	44	264	469	57.8
1	2	0	$8.92363 \cdot 10^{-17}$	1	1	44	208	469	52.8

The number of correct decimal digits of the approximation, calculated for $n_{it} = n_d = 1$, varies between 12 and 15:

$$\begin{aligned}
\widetilde{a}_1 &= 130.8900000000426\\
\widetilde{a}_2 &= 52.59999999998969\\
\widetilde{u}_1 &= 6.730000000000008\\
\widetilde{u}_2 &= 9.342000000000636\\
\widetilde{s}_1 &= 1.199999999999803\\
\widetilde{s}_2 &= 0.9700000000003587
\end{aligned}$$

For other values of n_{it} and n_d the approximations are similar. In the case $n_{it} = n_d = 1$, the time needed on a SUN SparcStation 1 is about 3 seconds.

If we apply our method by using inclusion functions of the derivatives, we get for f^*, x^* the bounds

$$\begin{aligned}
f^* &\subseteq [0, 3.6781148048291 97e \cdot 10^{-21}]\\
x^* &\subseteq \begin{pmatrix} [130.8899999999997, 130.8900000000002]\\ [52.59999999999987, 52.60000000000012]\\ [6.729999999999998, 6.730000000000003]\\ [9.341999999999995, 9.342000000000006]\\ [1.199999999999996, 1.200000000000004]\\ [0.9699999999999923, 0.9700000000000075] \end{pmatrix}
\end{aligned}$$

with the computational costs

n_{it}	n_d	n_M	n_L	l_S	n_{rf}	n_{if}	n_{ig}	n_{ih}	t_{STU}
≥ 14	1	1	1	153	487	19129	1	14	1688.00
≥ 7	2	1	1	164	270	19129	1	14	1681.65
≥ 5	3	1	1	142	296	19135	1	14	1681.90
≥ 4	4	1	1	151	295	19147	1	14	1687.35

Therefore, we need about $300s$ to calculate the above inclusions.

Example 3: In system analysis, a commonly occuring problem is to minimize the maximal real part of the eigenvalues of a matrix in order to get a maximally stable system. The systems discussed here consist of a matrix $M(x) \in \mathrm{I\!R}^{m\times m}$ with parameters $x \in \mathrm{I\!R}^2$. If $\lambda_i(x)$ are the eigenvalues of $M(x)$ and $f(x) := \max_i \Re(\lambda_i(x))$, then the aim is

$$\min_{x\in X} f(x).$$

We consider the matrix

$$M(x) = \begin{pmatrix} d_1(x_1,x_2) & k\sin x_1 & k\sin x_2 & k\cos x_1 & k\cos x_2 \\ k\sin 2x_1 & d_2(x_1,x_2) & kx_1 & kx_2 & kx_1x_2 \\ k\sin 2x_2 & k(x_1+x_2) & d_3(x_1,x_2) & kx_1^2 & kx_2^2 \\ k\cos 2x_1 & k(x_1-x_2) & k(x_1+x_2)^2 & d_4(x_1,x_2) & k\sin x_1x_2 \\ k\cos 2x_2 & kx_1x_2^2 & k4x_2^2 & k\sin(x_1+x_2) & d_5(x_1,x_2) \end{pmatrix},$$

with

$$\begin{aligned} d_1(x_1,x_2) &= 17.5 - 2e^{-500\left((x_1+4)^2+(x_2+4)^2\right)} - \frac{x_1+x_2}{20}, \\ d_2(x_1,x_2) &= 20 - \frac{x_1^2+x_2^2}{7} - (x_2+5)\cos\frac{\pi}{2}\left(x_1^2+x_2^2\right), \\ d_3(x_1,x_2) &= 20 - 6\cos 2\pi x_1, \\ d_4(x_1,x_2) &= 18 - \frac{x_1^4+x_2^4}{128} + \frac{1}{2}\cos 6\pi x_1x_2, \\ d_5(x_1,x_2) &= 20 - 6\cos 2\pi x_2, \\ k &= 10^{-3}, \\ x_1, x_2 &\in [-5,5]. \end{aligned}$$

As it can be seen in the plot of $-f(x)$ (we turned it upside down to let the global minimum be visible as global maximum), the function $f(x)$ contains lots of local minima and maxima.

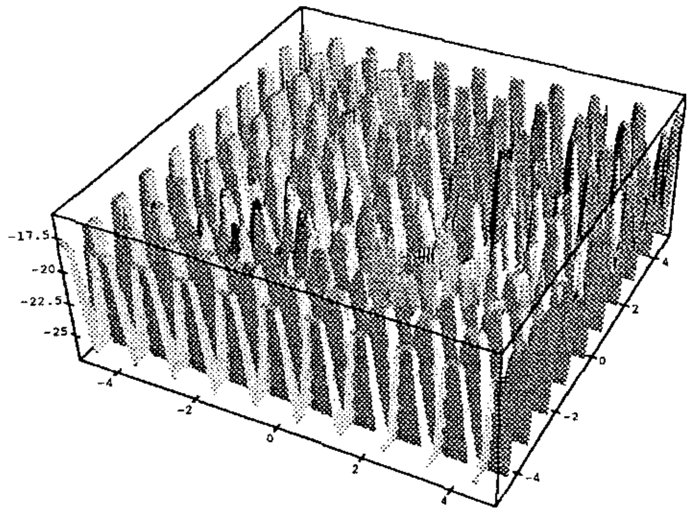


Figure 4.3: Plot of $f(x) := \max_i \Re(\lambda_i(x))$.

The unique global minimum is

$$f^* = 15.9, \quad x^* = (-3.99997, -3.99997)$$

Applying our method, we obtain the results

n_{it}	n_{d}	$\underline{F}^*$	$\overline{F}^*$	n_{M}	n_{L}	l_{s}	n_{rf}	n_{if}	t_{STU}
2	2	15.8686	17.1890	–	2	44	815	114	144.3
2	4	15.8974	15.9000	1	1	1	133	33	21.5
3	4	15.8999	15.9000	1	1	1	134	49	28.5

For almost all test problems known to us (cf.[14], [15]), our branch and bound method works very efficiently. Only in three cases we were not so lucky. One of them was Mandel'shtam's problem (cf. [6]), where we had success only up to dimension 4. This was because we found no appropriate inclusion function.

The efficiency depends very much on choosing an appropriate inclusion function. In many cases natural interval evaluations are sufficient. But we think that it is very important to find other types of inclusion functions for special problems.

Acknowledgement: The authors wish to thank Prof. Dr. H. Brakhage for initiating the research on self-validating methods.

Literatur

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions.* Dover publications, New York, 1968.

[2] G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, New York, 1983.

[3] H. Becker and R. Wittmer. Parameteroptimierung. In *Texte des Modellversuchs zur mathematischen Weiterbildung*, volume Heft 1. Universität Kaiserslautern, 1983.

[4] H. Brakhage. Über die numerische Behandlung von Integralgleichungen nach der Quadraturformelmethode. *Numerische Mathematik 2*, pages 183–196, 1960.

[5] K.D. Braune. *Hochgenaue Standardfunktionen für reelle und komplexe Punkte und Intervalle in beliebigen Gleitpunktrastern.* Dissertation, Universität Karlsruhe, 1987.

[6] V.F. Dem'yanov and V.N. Malozemov. *Introduction to MINIMAX.* John Wiley & Sons, 1974.

[7] L.C.W. Dixon and G.P. Szegö (eds.). *Towards Global Optimization.* North-Holland, Amsterdam, 1975.

[8] G. Golub and C. van Loan. *Matrix Computations.* John Hopkins University Press, second edition, 1989.

[9] A. Griewank. On Automatic Differentiation. In *Mathematical Programming 88.* Kluwer Academic Publishers, Boston, 1989.

[10] E.R. Hansen. *Global Optimization using Interval Analysis.* Marcel Dekker, New York, Basel, Hong Kong, 1992.

[11] R. Horst and H. Tuy. *Global Optimization.* Springer-Verlag, Berlin, 1990.

[12] IEEE Standard for Binary Floting-Point Arithmetic. *ANSI/IEEE Standard 754*, 1985.

[13] C. Jansson. On Self-Validating Methods for Optimization Problems. In J. Herzberger, editor, *Topics in Validated Computations — Studies in Computational Mathematics*, Amsterdam. North-Holland to appear.

[14] C. Jansson and O. Knüppel. A Global Minimization Method: The Multidimensional case. Technical Report 92.1, Forschungsschwerpunkt Informations- und Kommunikationstechnik, TU Hamburg-Harburg, 1992.

[15] C. Jansson and O. Knüppel. Numerical results for a Self-Validating Globa Optimization Method. Technical Report 94.1, Forschungsschwerpunkt Informations- und Kommunikationstechnik, TU Hamburg-Harburg, 1994.

[16] O. Knüppel. PROFIL — Programmer's Runtime Optimized Fast Interval Library. Technical Report 93.4, Berichte des Forschungsschwerpunktes Informations- und Kommunikationstechnik, TUHH, 1993.

[17] W. Krämer. *Inverse Standardfunktionen für reelle und komplexe Intervallargumente mit a priori Fehlerabschätzung für beliebige Datenformate.* Dissertation, Universität Karlsruhe, 1987.

[18] R. Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing 4*, pages 187–201, 1969.

[19] R. Lohner. Enclosing all Eigenvalues for Symmetric Matrices. In Ch. Ullrich and J. Wolff von Gudenberg, editors, *Accurate Numerical Algorithms*, New York, 1989.

[20] MATLAB User's Guide. *The MathWorks Inc.*, 1987.

[21] R. Moore, E. Hansen, and A. Leclerc. Rigorous Methods for Global Optimization. In *In Recent Advances in Global Optimization, Princeton series in computer science*, pages 321–342, Princeton, New Jersey, 1992. Princeton University Press.

[22] R.E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1966.

[23] R.E. Moore. A Test for Existence of Solutions for Non-Linear Systems. *SIAM J. Numer. Anal. 4*, 1977.

[24] A. Neumaier. *Interval Methods for Systems of Equations, Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1990.

[25] P.M. Pardalos and J.B. Rosen. Constrained Global Optimization: Algorithms and Applications. *Springer Lecture Notes Comp. Sci. 268, Berlin*, 1987.

[26] M. Plum. Numerical Existence Proofs and Explicit Bounds for Solutions of Nonlinear Elliptic Boundary Value Problems. *Computing*, 49:25–44, 1992.

[27] L.B. Rall. Automatic Differentiation: Techniques and Applications. In *Lecture Notes in Computer Science 120.* Springer Verlag, Berlin-Heidelberg-New York, 1981.

[28] H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization.* John Wiley & Sons (Ellis Horwood Limited), New York (Chichester), 1988.

[29] S.M. Rump. Verification Methods for Dense and Sparse Systems of Equations. In J. Herzberger, editor, *Topics in Validated Computations — Studies in Computational Mathematics*, Amsterdam. North-Holland to appear.

[30] S.M. Rump. Validated Solution of Large Linear Systems. In R. Albrecht, G. Alefeld, and H.J. Stetter, editors, *Computing Supplementum 9, Validation Numerics*, pages 191–212. Springer, 1993.

[31] K. Strubecker. *Einführung in die höhere Mathematik.* Oldenbourg Verlag, München, 1967.

[32] A. Törn and A. Žilinskas. *Global Optimization.* Springer-Verlag, Berlin, Heidelberg, New York, 1989.

[33] J.H. Wilkinson. Modern Error Analysis. *SIAM Rev. 13*, pages 548–568, 1971.