ITHMETIC AND ILL-CONDITIONED ALGEBRAIC PROBLEMS†

Günter Schumacher

Universität Karlsruhe
Institut für Angewandte Mathematik
Kaiserstr. 12, D-7500 Karlsruhe
West-Germany

Abstract

Interval arithmetic, omputation with numbers which are aff tolerances, always provides reliable when applied in numerical algorithms rs. It guarantees that the exact result ithm lies within the computed tolerance ill-conditioned cases these bounds ma ly wide and although the statement s valid, it is in practice worthless. ich have been recently introduced a rovide the possibility of successi ing the tolerance. Furthermore, th nd uniqueness of the solution is prove ematical sense) by the computer.

These methods combin pts of interval analysis with the con etic defined by Kulisch and Miranker. based on fixed point theorems and lar product is essential for their i .

0. Introduction

In [8] a mathematica as been laid of what we call "Compu ". For example, the algebraic structu tal computer may be described in ter reen", and the process of truncati number to its machine representati o be a mapping from the real numbe en of floating-point numbers.

We will summarize som d concepts:

$\mathbb{R}$ ..... set of re
S ..... set of t numbers of a given di
$V_n T$ ..... vector n over a set $T \in \{\mathbb{R}, S$
$M_n T$ ..... square r rows and components i
$\mathbb{P}T$ ..... power se $\mathbb{R}, V_n S, M_n \mathbb{R}, M_n S\}$
IT ..... set of an ordered set $T \in \{\mathbb{R}, S, V$ }, i.e. the set of all $[a$ $\mid a \leq x \leq b\}$ (For s properties of interval see [1],[6]).

†This paper is an e from [4].

Occasionally, we will denote a member of an interval A by $\underset{\bullet}{A}$.

For two intervals $A,B \in T$ an arithmetic operation $\circ \in \{+,-,\cdot,/\}$ is defined by
$A \circ B := \{a \circ b \mid a \in A, b \in B\}$

$\cup$ ..... convex union operation for two intervals $A,B \in IS$:
$A \cup B := \{x \in S \mid \exists\, a \in A, b \in B : a \leq x \leq b \ \vee \ b \leq x \leq a\}$

$d(\cdot)$ ..... diameter of an interval, i.e.
$A = [a,b] \in IS \implies d(A) := b-a$

$m(\cdot)$ ..... midpoint of an interval
(in our context m(A) may take any point out of an interval $A \in IS$)

$\Diamond : \mathbb{P}\mathbb{R} \to IS$ rounding from $\mathbb{P}\mathbb{R}$ into IS with the property

$$\bigwedge_{A \in \mathbb{P}\mathbb{R}} \Diamond A = \bigcap \{X \in IS \mid A \subset X\}$$

For vectors and matrices this definition applies componentwise

$\diamondsuit, \diamondsuit, \ldots$ interval operations in IS, IV_nS and IM_nS
$\diamondsuit, \diamondsuit$ defined for $T \in \{S, V_nS, M_nS\}$ by

$$\bigwedge_{A,B \in IT} \ \bigwedge_{\circ \in \{+,-,\cdot,/\}} A \diamondsuit B := \Diamond(A \circ B)$$

Throughout this paper we consider the following problem:

"Let $f : D \to V_n\mathbb{R}$ be a continuously differentiable function, $D \subset V_n\mathbb{R}$. We seek an $\hat{x} \in V_n\mathbb{R}$ with $f(\hat{x})=0$, or more practically, we ask for an inclusion $X \in IV_n\mathbb{R}$ of $\hat{x}$, where the diameter d(X) is sufficiently small."

We restrict ourselves to the case where the components of f are arithmetic expression with operations $+,-,\cdot,/$ and integer exponentiation. The algorithm to be introduced determines an inclusion X of the solution $\hat{x}$ and performs an automatic verification of conditions such as the existence and uniqueness of a zero $\hat{x}$ of f within X. These methods are called E-methods (E is the first letter of the three German words "Existenz" for existence, "Eindeutigkeit" for uniqueness, and "Einschließung" for inclusion.

CH2419-0/87/0000/02 IEEE

For this purpose, the problem of finding a zero $\hat{x}$ of f is transformed into a fixed point equation $g(x) = x$ (see [7], [11], [12]). An inclusion of the solution of this fixed-point problem is computed iteratively in the space $IV_n\mathbb{R}$ starting with an approximate solution $\tilde{x} \in V_n\mathbb{R}$. By use of residual correction techniques the diameter of the resulting interval is diminished more and more.

1. Theoretical foundations

We summarize some theorems from [7], [11], [12] concerning the existence and uniqueness of a solution of a system of equations. At first, we need the following

Lemma 1: (Schauder's fixed point theorem)

Let $f : X \to V_n\mathbb{R}$ be a continuous function on $X \subseteq V_n\mathbb{R}$, X nonempty, convex and compact. If $f(X) \subseteq X$, then the equation $f(x) = x$ has at least one solution $\hat{x}$ in X.

Proof: see [5], e.g. □

From this lemma we have immediately

Theorem 1: Let $f : X \to V_n\mathbb{R}$ be a continuous function on $X \in IV_n\mathbb{R}$, and $F : \mathbb{P}X \to \mathbb{P}V_n\mathbb{R}$ with

$$\bigwedge_{A \in \mathbb{P}X} \bigwedge_{x \in A} f(x) \in F(A). \quad (1.1)$$

If $F(X) \subseteq X$, then the equation $f(x)=x$ has at least one solution $\hat{x}$ in X and

$$\bigwedge_{k \geq 0} \hat{x} \in F^k(X) ,$$

where $F^0(X) := X$, $F^k(X) := F(F^{k-1}(X))$.

Proof: From $X \in IV_n\mathbb{R}$ we have: $X \neq \phi$, X compact and convex. From the definition we have $f(X) := \{f(x) \mid x \in X\} \subseteq F(X) \subseteq X$. Therefore, Lemma 1 delivers the existence of a solution $\hat{x} \in X$ of $f(x) = x$. The rest is shown by induction. □

F may be arbitrarily chosen to satisfy (1.1). In practical applications F will be any interval extension of f.
Theorem 1 allows no conclusion about the uniqueness of the solution. For this purpose we define

$$\bigwedge_{A,B \in IR} A \subsetneq B :\Leftrightarrow A \subseteq B \wedge A \neq B$$

We also need the following two lemmata from [15]:

Lemma 2: Let $A = ((a_{ij})) \in M_n\mathbb{R}$, $x = (x_i) \in V_n\mathbb{R}$ and $x>0$. Then for the spectral radius,

$$\max_{1 \leq i \leq n} \frac{\sum_{j=1}^{n} |a_{ij}| x_j}{x_i}$$

ho

Lemma 3: Le : and $0 \leq |B| \leq A$. Then, $\rho(A)$.

From this we

Theorem 2: Le $C \in IM_n\mathbb{R}$, a

$$\wedge \not\subseteq X , \quad (1.2)$$

wh s defined componentwise.
Th

$(C) < 1$.

Proof: Ob $(X) > 0$. Furthermore, we h $\cdot X) = d(Z + C \cdot X) < d(X)$ a

$d(X)$.
Fi ve $d(C \cdot X) \geq |C| \cdot d(X)$
$\Rightarrow$ $< d(X)$. (1.3)
Le $d(X)$. Then by lemma 2

$$\frac{y_i}{d(X_i)} \overset{(1.3)}{<} \frac{d(X_i)}{d(X_i)} = 1 .$$

Th $\in C$ we have

$$\rho(|C|) \leq \rho(|C|) < 1$$

by emma 3 twice. □

Now we can pr llowing theorem which is a constructive ify the existence and the uniqueness of ion of a system of equations.

Theorem 3: Le $X \in V_n\mathbb{R}$, $f : X \cup \tilde{x} \to V_n\mathbb{R}$, f). For arbitrary but fixed Y let $J(Y) \in IM_n\mathbb{R}$ and assume th very set of n vectors y_1

$$\left.\begin{matrix} \cdots & \frac{\partial f_1}{\partial x_n}(y_1) \\ & \vdots \\ \cdots & \frac{\partial f_n}{\partial x_n}(y_n) \end{matrix}\right) \in J(Y)$$

ho $: \mathbb{P}X \to \mathbb{P}V_n\mathbb{R}$ be defined by
$G($ $(\tilde{x}) + (I - R \cdot J(\tilde{x} \cup Y)) \cdot (Y - \tilde{x})$
fo X, where $R \in M_n\mathbb{R}$ is an
arl fixed matrix.

If

$$G(\quad) \qquad (1.4)$$

holds, [illegible] equation $f(x) = 0$ has one and [illegible] solution $\hat{x}$ in X and $\hat{x} \in \mathbb{G}^k$.

Proof: Consider [illegible] tinuous) function $g : X \cup x$ [illegible] $g(x) = -R \cdot f(x)$. From the [illegible] alue theorem we have that for [illegible] $\subset X$ and $y \in Y$ there exists a [illegible] $(\tilde{x} \cup Y)$ such that

$$g(y) \quad (\tilde{x}) + M(Y)\cdot(y-\tilde{x}))$$
$$\tilde{x}) + (I-R\cdot M(y))\cdot (y-\tilde{x})$$
$$\in \quad .$$

Theor[illegible].4) then delivers the existe[illegible] fixed point $\hat{x}$ of f. Furthe[illegible] (1.4) we have

$$-Rf(\tilde{\ }) \quad (\tilde{x} \cup X))\cdot(X-\tilde{x}) \subsetneq X-\tilde{x}$$

Hence, [illegible] implies $\rho(I-R\cdot M) < 1$ for all [illegible] $\cup X)$. When considering the eig[illegible] of $I-R\cdot M$ we see that R and M [illegible]r. With $\hat{x}-R\cdot f(\hat{x}) = \hat{x}$, it foll[illegible] $f(\hat{x}) = 0$. Let $\hat{y} \in X$ be a fur[illegible]tion. The generalized mean[illegible] rem implies the existence [illegible] $\in J(\tilde{x} \cup X)$ with

$$) + M(\hat{y})\cdot(\hat{y} - \hat{x}) \quad .$$

But M[illegible] ar and hence $\hat{y} = \hat{x}$. □

In practical ap[illegible] ie matrix R will be an approximation o[illegible] e of the Jacobian $J(\tilde{x})$. Theorem 3 has be[illegible] ed for application on computers. In [illegible] pproach the goal of computer applic[illegible] st observation is that for any G^*: $\tilde{x}$ [illegible] ch satisfies

$$Y \subset \tilde{\ } \qquad \subset G^*(Y)$$

it is obvious th[illegible] $\subsetneq Y$ implies $G(Y) \subsetneq Y$.

This fact all[illegible] ert that theorem 3 remains valid i[illegible] ed by S and G(Y) by an interval arit[illegible] on of G, i.e.

$$G^*(Y) := \quad (\tilde{x})$$
$$\cup Y)) \diamondsuit (Y \diamondsuit \diamondsuit\tilde{x}) \quad .$$

$\diamondsuit f$ denotes any [illegible] f with

$$\bigwedge_{x \in V_n\mathbb{R}} f$$

The definitio[illegible] r product

$$a \diamondsuit b := \diamondsuit(a$$

requires an exact summation algorithm as stated in [8]. With such a tool the critical residual term $\diamondsuit(I - R \cdot J(\tilde{x} \cup Y))$ may be evaluated with only one rounding in each component.

Next, a theorem can be formulated directly applicable on computers. For a better understanding of the following corollary it should be remarked that, in practical application, the inclusion of the absolute error $\hat{x} - \tilde{x}$ of an approximate solution $\tilde{x}$ leads to better results than an inclusion for $\hat{x}$ itself. This is summarized in the following

Corollary: Let $\tilde{x} \in V_nS$, $Z \in IV_nS$, and $f \in C^1((\tilde{x} \diamond Z) \cup \tilde{x})$. For every arbitrary but fixed $Y \subset (\tilde{x} \diamond Z) \cup \tilde{x}$, let $J(Y) \in IM_nS$ and assume that for every set of n vectors $y_1, \ldots, y_n \in Y$,

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1}(y_1) & \cdots & \frac{\partial f_1}{\partial x_n}(y_1) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(y_n) & \cdots & \frac{\partial f_n}{\partial x_n}(y_n) \end{pmatrix} \in J(Y)$$

holds. Let $H : \mathbb{P}Z \to IV_nS$ be defined as

$$H(Y) := -R \diamond \diamondsuit f(\tilde{x}) \diamond \diamondsuit(I-R\cdot J(\tilde{x} \cup (\tilde{x} \diamond \diamondsuit Y))) \diamond \diamondsuit Y \qquad (1.5)$$

for $Y \in \mathbb{P}Z$, where $R \in M_nS$ is an arbitrary but fixed matrix. If

$$H(Z) \subsetneq Z \ , \qquad (1.6)$$

then the equation $f(x) = 0$ has one and only one solution $\hat{x}$ in $\tilde{x} \diamond Z$ and $\hat{x} - \tilde{x} \in H^{(k)}(Z)$, $k \geq 0$.

2. Equation solving with error control

The corollary provides the second of the two basic steps in computing verified bounds for the solution of a system of equations:

1) an approximation step to determine a sufficiently good estimate $\tilde{x}$.
2) an inclusion step to determine the enclosure Z of the absolute error $\hat{x} - \tilde{x}$ with respect to $\tilde{x}$.

In this chapter we will discuss some implementation aspects of these two points. Let us start with the inclusion step.

2.1 Implementation of the inclusion step

According to [11] we define an ϵ-inflation for an interval $A \in IS$ by

$$A \circ \epsilon := \begin{cases} A + [-1, 1] \cdot \epsilon \cdot d(A) & \text{for } d(A) \neq 0 \\ A + [-\eta, +\eta] & \text{for } d(A) = 0 \end{cases}$$

Here η is the smallest positive floating-point number of the computer in use. For interval vectors this definition applies componentwise. ϵ-inflation is indispensable to assure the convergence of the interval iteration. In practice, 0.1 turned out to be a good value for ϵ.

Using the corollary, a verified inclusion may be obtained by performing the following iteration in the space IV_nS:

```
Y := 0 ;  count := 0 ;

repeat

   count := count + 1 ;
   Z := Y ∘ ε;
   Y := -R⊛◇f(x̃) ⊛
                ◇(I-R•J(x̃ ∪ (x̃⊛Z)))⊛Z

until ( Y ⊊ Z ) or (count = count_max);
```

If the termination criterion $Y \subsetneq Z$ is satisfied, then by corollary 1 it is verified that the exact solution $\hat{x}$ lies in $\tilde{x} \circledast Z$ and that no other solution lies between these bounds. (The integer variable count is used to prevent infinite looping in cases where the iteration is not convergent).

2.2 Performing the approximation step

There are a large number of different methods for obtaining an approximation for $\hat{x}$ (see e.g. [9]). In the following any of those principles may be used. Here we describe only the "classical" Newton's method

$$x^{(k+1)} := x^{(k)} - (f'(x^{(k)}))^{-1} \cdot f(x^{(k)}) \qquad (2.6)$$

since the problem of evaluating the function f at a point $x^{(k)}$ with high accuracy is common to all methods. If it is not possible to compute the residual term $f(x^{(k)})$ accurately, we have no measure how near we are to the exact solution.

As stated above, the components f_i of f are considered as arithmetic expressions. The computation of $f_i(x)$ is usually done on a computer step by step, i.e. using an algorithm which performs one operation in the formula after the other. The results of the operations are stored in some intermediate variables $z_1, \ldots, z_n$, say.

Example: The

$\cdot(x_3+x_4)/(x_5-x_6)$

wi

2

1

2

3

1

usi e results $z_1, \ldots, z_5$.

We will state way to arrive at a highly accurate the function values:

The computatio be considered as a successive eva system of nonlinear equations for $z_1, \ldots, z_n$. This system of equations ha form. Hence, for the computation of of $z_n = f_i(x)$ with a guaranteed acc nsider the following special type o em:

$$g_1(z$$
$$g_2(z$$
$$\cdot$$
$$\cdot$$
$$\cdot$$
$$g_n(z$$

(In [2] and [3] nlinear system was transformed into a s linear equations by using term rewriting es. The resulting system also has trian n and can be solved with the principles

We assume g_k ntiable in terms of $z_1, \ldots, z_k$. Let $\hat{z}$ the approximations and $\hat{z}_1, \ldots, \hat{z}_n$ the e .

We define

$$\Delta z_j := \hat{z}_j \quad , \quad j = 1, \ldots, n$$

and seek an inc e correction

$$\Delta z_j \in \Delta \quad , \quad j = 1, \ldots, n$$

Applying the m orem to g_k with respect to the last va in the interval $\hat{z}_k \cup \tilde{z}_k$ yields

$g_k(\hat{z}_1, \ldots, \hat{z}_k) = {}_k(\hat{z}$) +

$(\hat{z}_1, \hat{z}_2, \ldots, \zeta_k) \cdot \Delta z_k$,

with $\zeta_k \in \hat{z}_k \cup \tilde{z}_k$

Continuing thi pr ... z_{k-1}, z_{k-2}, ...

results in

$g_k(\hat{z}_1, \ldots, \hat{z}_k) = {}_k(\hat{z}$ +

$+ \sum_{j=1}^{k} \frac{\partial}{\partial z_j} {}_k(\ldots, \zeta_j, \tilde{z}_{j+1}, \ldots, \tilde{z}_k) \cdot \Delta z_j$

$\hat{z}_j \cup \tilde{z}_j$, $j = 1, \ldots, k$

Since $\hat{z}_1, \ldots,$ ... act solution we have

$g_k(\hat{z}_1, \ldots, \hat{z}_k) \quad 0$. e have

$\Delta z_k = \Big[- g_k(\tilde{z} \ldots$

$- \sum_{j=1}^{k-1} \frac{\partial}{\partial z_j} \ldots \tilde{z}_{j+1}, \ldots, \tilde{z}_k) \cdot \Delta z_j \Big] /$

$\frac{\partial}{\partial z_k} g_k(\hat{z}_1, \ldots, \hat{z}_{k-1}, \zeta_k)$

under the as ... ot ... $\frac{\partial}{\partial z_k} g_k(\hat{z}_1, \ldots, \hat{z}_{k-1}, \zeta_k) \neq 0$.

Via induction we ... show

$\Delta z_k \in \Delta Z_k := -g$) +

$- \sum_{j=1}^{k-1} \frac{\partial}{\partial z_j} g_k$ ... $\Delta Z_j) \tilde{z}_j, \tilde{z}_{j+1}, \ldots) \cdot \Delta Z_j \Big] /$

$\frac{\partial}{\partial z}$ ... $\tilde{z}_{k-1} + \Delta Z_{k-1}, (\tilde{z}_k + \Delta Z_k) \cup \tilde{z}_k)$

We must a ... de minator does not contain zer ... Th ... the well-known condition from th ... im ... ction theorem that

$\frac{\partial}{\partial z_k} g_k$ ...)

to guarant ... ty of $g_k = 0$ for z_k in some neig ... i: ... $(\hat{z}_1, \ldots, \hat{z}_k)$.

It is necc ... k th ... following:

- To u ... rm ... to compute bounds for ... ime ... hat $\frac{\partial}{\partial z_k} g_k$ does not depen ... is always true for all kind ... arising from arithmetic expr ...] and the example above).
- The ... ${}_k(\tilde{z}, \ldots, \tilde{z}_k)$ must be of high accuracy to guarantee a sufficiently sharp inclusion. This is also possible for systems arising from arithmetic expressions by applying the exact scalar product as developed in [8]. For this purpose a "division equation" $z_k = z_i / z_j$ has to be transformed into $z_k \cdot z_j = z_i$.

In most cases it is not necessary to compute the value of the partial derivatives with the same effort as the function value since the success of (2.6) mainly depends on the accuracy of $f(x^{(k)})$. Nevertheless, it is also possible to determine the partial derivatives with comparable accuracy. (For the technique of computing the partial derivatives of a function we refer to [10]).

With the methods for linear systems from [7], each Newton step may be performed with high accuracy since each step is the solution of a system of linear equations

$$f'(x^{(k)}) \cdot (x^{(k)} - x^{(k+1)}) = f(x^{(k)})$$

The methods from [7] are specializations of the method described here, and so this way of performing a Newton step may be considered as a recursive call of the procedure in this paper. The possibility to do that is important in the cases where the Jacobian is ill-conditioned.

2.3 An algorithm for the solution of a system of nonlinear equation with maximum accuracy

We use the following notations: $\hat{x}$ denotes the exact solution of the system of nonlinear equations $f(x) = 0$, b is the base of the floating-point system in use, t the length of the mantissa.

The approximation of $\hat{x}$ in the k-th step is $\sum_{i=o}^{k} \tilde{x}^{(i)}$ where $\tilde{x}^{(i)} \in S$, $i=1,\ldots,n$. This form of our approximation $\tilde{x}$ is usually called a "staggered correction form" of $\tilde{x}$ since all values $\tilde{x}^{(i)}$ are actually stored and the summation will be done when we have finished. To this approximation we compute an inclusion Z of the next correction

$$x^{(k)} := \hat{x} - \sum_{i=o}^{k-1} \tilde{x}^{(i)} .$$

In step (N3) of this algorithm, least bit accuracy is forced. This may be weakened if t is replaced by some positive integer $t_o < t$. In case one of the components of the exact solution is zero, some refined termination criterion may replace the one stated here. This should be done since the computation of an inclusion $[-\eta, \eta]$ of zero is rather expensive.

Algorithm:

```
        k := 0 ;                    { correction counter }
(N1)    Solve f( Σ_{i=0}^{k-1} x̃(i) + x(k) ) = 0
        approximately for the unknown x(k)
        using Newton's method (2.6) with
        formula evaluation technique; in ill-
        conditioned cases perform each step
        with high accuracy using the linear
        techniques from [7]; call the appro-
        ximation x̃(k);

(N2)    Compute an approximation R of
        f'( Σ_{i=0}^{k} x̃(i) )^-1 ;
        j := 0 ;  { iteration counter }
        Z := Y ;  X := ◇( Σ_{i=0}^{k} x̃(i) ) ;
        repeat
          j := j+1;
          Z := Y ∘ ε;
          Y := W ⊛ ◇(I-R•J(X ∪ (X⊛Z)))⊛Z ;
          success := (Y ⊊ Z) ;
        until success or (j = 10);

(N3)    if success
        then dm := d(X⊛Z) ;
             if dm < |X⊛Z| • b^-t+1 goto (N4)
        else
             k := k+1 ;
             if k > 10
             then write('No solution
                         achieved'); stop
             else goto (N1)

(N4)    Result: x̂ ∈ X⊛Z
```

3. Examples of application

The following two examples demonstrate the typical behaviour of the equation solver presented above. They will show the need of an exact scalar product and, furthermore, a formula evaluation technique using this scalar product.

Example 1:

We define the following system of equations for an unknown vector $x \in V_n\mathbb{R}$:

$$f(x) := \alpha(H \cdot x + \psi(x)) = 0$$

where $H \in M_n\mathbb{R}$ is the Hilbert matrix of degree n, i.e. $H = ((h_{ij}))$, $h_{ij} := 1/(i+j-1)$, $i,j = 1,\ldots,n$. $\psi : V_n\mathbb{R} \to V_n\mathbb{R}$ is defined by

$\psi(x) := ($ $= x_i +$ $_i$, $\epsilon_i > 0$, i=1,...,n

The whole
multiplic
an exact
easily s
solution.
ϵ_i, the
the solu
teristic
68000 mi
decimal
the ϵ_i
values,
ponent x_i
Newton s
4 answer
mation i
accuracy.
sion ste
always th

scal v the lowest common
mpo s of H to guarantee
on e data. It can be
e p n has exactly one
, fo reasing n and small
is tm is almost H since
bec s small. The charac-
ble where computed on a
PAS C using a 13 digit
all mples the values of
an mple for all other
y t result for the com-
3 have the number of
N1) he algorithm. Column
wh the Newton approxi-
is t performed with high
spla s the number of inclu-
he initial value for x was

n	in th		pr. eps	high acc.?	incl. steps
5	-5.000	-12	2	no	2
6	€	-1	3	no	2
7	-6.	-12	3	no	2
8	7.	-12	4	no	2
9	-1.	-11	8	yes	2
10	-3.	-12	13	yes	2
11	-1.834	-04	27	yes	2
12	-2.	-03	34	yes	2
13	8.	-0	27	yes	1
14	4.	-06	95	yes	1
15	1.	-06	35	yes	2

ble

Example 2:

The follow
engineeri
the so-call

$x_0 =$
$x_i =$ $1 - x$ $_{-1})$, i = 1,...,n.

m rises from control
c uits and is known as
n e ation":

The value
an expli
Nevertheless,

theo ically determined by
n of the right-hand side.
the est arithmetic will

approximate
Table 3.2a

exact scalar
we apply tha
by simple r
underlined.
4361, the v
the ACRITH
results of
of table 3.

places per step.
ts for x_n without

n table 3.2b when
e of evalu- ation
correct digits are
s done on an IBM
e obtained by using
al values for the
values of column 2
= 0.3, λ = 0.95.

n	r	on by terval arithmetic
10	0.9	0.9^{12}_{08}
20	0.9	0.9484^{8}_{4}
30	0.9	43.3
40	0.6	-31.2
50	0.7	"overflow"
60	0.	----
70	0.	----
80	0.	----
90	0	----
100	0	----

n	r system evaluation with exact scalar product
10	$0.933608886103744^{5}_{4}$
20	$0.9484\ 1664586818^{9}_{8}$
30	$0.9497\ 6664490856^{5}_{4}$
40	$6148\ 7509460460^{2}_{1}$
50	$7722\ 3942734024^{8}_{1}$
60	$0.334\ 8960225076^{1}_{0}$
70	1258545544^{8}_{6}
80	$5973\ 0842574404^{3}_{2}$
90	$2109\ 1469055217^{8}_{7}$
100	$0.5866\ 8531833946^{1}_{0}$

References:

[1] Alefeld, G., Herzberger, J.: Einführung in die Intervallrechnung, Bibl. Inst., Mannheim-Wien-Zürich (1974)

[2] Böhm, H.: Berechnung von Polynomnullstellen und Auswertung arithmetischer Ausdrücke mit garantierter maximaler Genauigkeit, Doktor-Dissertation, Universität Karlsruhe (1983)

[3] Böhm, H., Rump, S.M.: Least Significant Bit Evaluation of Arithmetic Expressions in Single-Precision, Computing 30, 189-199 (1983)

[4] Böhm, H., Rump, S.M., Schumacher, G.: E-Methods for Nonlinear Problems, to appear in: "Computer Arithmetic: Scientific Computation and Programming Languages", Teubner Verlag, Stuttgart (1987)

[5] Heuser, H.: Funktionalanalysis, 2nd ed., Teubner Verlag, Stuttgart (1986)

[6] Kaucher, E.: Interval Analysis in the Extended Interval Space IR, Computing Suppl. 2, 33-49 (1980)

[7] Kaucher, E., Rump, S.M.: E-Methods for Fixed Point Equations f(x)=x, Computing 28, 31-42 (1982)

[8] Kulisch, U., Miranker, W.: Computer Arithmetic in Theory and Practice, Academic Press, New York (1981)

[9] Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970

[10] Rall, L.B.: Automatic Differentiation, Techniques and Applications. Lecture Notes in Computer Science, No 12, Springer, Berlin (1981)

[11] Rump, S.M.: Solving nonlinear systems with least significant bit accuracy, Computing 29, 183-200 (1982)

[12] Rump, S.M.: Solving algebraic problems with high accuracy in: "A new Approach to Scientific Computation", p. 51-120, Academic Press, New York (1983)

[13] Stark, J.: Maximal genaue Auswertung arithmetischer Ausdrücke durch Lösen nichtlinearer Gleichungssysteme, Diplomarbeit at the Institut fur Angewandte Mathematik, Universität Karlsruhe (1985)

[14] Varga, R.S.: Matrix Iterative Analysis, Prentice Hall, Englewood Cliffs (1962)