SOLVING ALGEBRAIC PROBLEMS WITH HIGH ACCURACY

Siegfried M. Rump*

Institute for Applied Mathematics
University of Karlsruhe
Karlsruhe, West Germany

The paper gives a synopsis of new methods for solving algebraic problems with high accuracy. Examples of such problems are the solving of linear systems, eigenvalue/eigenvector determination, computing zeros of polynomials, sparse matrix problems, computation of the value of an arbitrary arithmetic expression (in particular the value of a polynomial at a point), non-linear systems, linear, quadratic and convex programming, etc. over the field of real or complex numbers as well as over the corresponding interval spaces.

We begin by demonstrating the effect of roundoff errors in numerical computation. We use several examples to show that in fact the error of a numerical computation may be arbitrarily large. Some of the examples can be performed on a pocket calculator. As a first step for avoidance of these errors we develop the fundamentals of a computer arithmetic including the precise dot product (Kulisch/Miranker theory).

Next we develop the fundamentals of our new methods. Every result given by an algorithm based on one of the new methods is automatically verified to be correct by the algorithm itself. A result includes an error bound. We say that the result is of high accuracy if the maximum relative error of each component is small. For this purpose we need a precise definition of the arithmetic of the computer in use.

All the algorithms based on our new methods have some key properties in common:

- every result is automatically verified to be correct by the algorithm
- the results are of high accuracy; the error of every component of the result is of the magnitude of the relative rounding error unit
- moreover the solution of the given problem is automatically shown to exist and to be unique within the given error bounds
- the computing time is of the same order as comparable (purely) floating-point algorithm (the latter, of course, offers none of the new features).

The key property of the algorithms is that error control is performed automatically by the computer without any requirement on the part of the user (such as estimating spectral radii. The efficiency of the algorithms will be shown, for instance, by inverting a Hilbert 15×15 matrix in a 12 decimal digit floating-point system. This (after multiplying with a proper factor) the Hilbert matrix of largest dimension which can be stored without rounding error in this floating-point system. The error bounds for all components of the inverse of the Hilbert 15×15 matrix are as small as possible, i.e., left and right bounds differ only by one in the 12^{th} place of the mantissa of each component. We call this least significant bit accuracy (lsba). Our experience shows that the results of the algorithms using our new methods very often have the lsba-property for every component of the solution.

Present address: IBM Deutschland, Entwicklung und Forschung, 7030 Boeblingen, West Germany

ISBN 0-12-428660-7

CONTENTS

INTRODUCTION

In this paper we deal with errors in numerical computations and discuss possibilities for their elimination. The problems we have in mind may consist of exactly representable data on a given computer ("point problems") or may be subjected to a certain error margin. Our aim is to give a solution with an error bound such that existence and uniqueness of the solution within these bounds is automatically verified. If this verification process fails a signaling message shall be given. Further the aim is to achieve least significant bit accuracy for point problems and smallest possible bounds for problems with uncertainties in the coefficients. The algorithms presented demonstrate that even for extremely ill-conditioned problems, such as inverting the Hilbert 21×21 matrix on a 14 hexadecimal digit computer, bounds with the least significant bit accuracy property can be found. The condition number of the 21×21 Hilbert matrix is approximately 10^{30}, and is the Hilbert matrix of largest dimension exactly storable on that computer.

To achieve this accuracy and, especially, to give only true results a precisely defined arithmetic is necessary. On a UNIVAC 1108 we have for instance

$$134217728.0 - 134217727.0 = 2.0$$

with exactly representable operands. Therefore, we define a computer arithmetic in the first chapter. The theoretical background and the required algorithms for certain classes of problems are given in the succeeding chapters. The algorithms have been implemented on a minicomputer based on Z80 with a 64 k Byte memory. There, a decimal arithmetic with 12 digits in the mantissa and a PASCAL-SC compiler is implemented. The minicomputer has been developed at the Institute for Applied Mathematics at the University of Karlsruhe and the Fachbereich Informatik of the University at Kaiserslautern (Professors Kulisch and Wippermann). Further, the algorithms are implemented on a UNIVAC 1108 and IBM 370/168. However, we cannot mention every detail of the implementation in the succeeding description of the algorithms. We give explicit algorithms for solving systems of linear equations and systems of nonlinear equations and the verification of the nonsingularity of a

(real or complex) matrix. Algorithms for the other problems discussed in the succeeding chapters can be derived easily from the stated theorems and corollaries as well as from the implementation hints for the explicitly given algorithms.

1. COMPUTER ARITHMETIC

Let T be one of the sets $\mathbb{R}$ (real Numbers), $V\mathbb{R}$ (real vectors with n components), $M\mathbb{R}$ (real $n\times n$ matrices), $\mathbb{C}$ (complex numbers), $V\mathbb{C}$ (complex vectors with n components) or $M\mathbb{C}$ (complex $n\times n$ matrices). Here and in the following the letter n is reserved to specify the length of a vector or the number of rows of a quadratic matrix. If the length of a vector is other than n this will be displayed in the corresponding set by an index (e.g. $V_{n+1}\mathbb{C}$). If the number of rows of a quadratic matrix is other than n we write, for instance, $M_{n+1}\mathbb{R}$, if the matrix is not quadratic this is made visible by two indices separated by a comma (e.g. $M_{\ell,m}\mathbb{R}$).

In the power set $I\!P T$, operations are defined by (with well-known restrictions for /)

$$A*B := \{a*b \mid a\in A \wedge b\in B\} \quad \text{for } A,B\in I\!P T,\ *\in\{+,-,\cdot,/\}.$$

Other operations such as $\cdot: I\!P M\mathbb{C}\times I\!P V\mathbb{R}\to I\!P V\mathbb{C}$ are defined similarly. The order relation $\leq$ in $\mathbb{R}$ is extended to $V\mathbb{R}$ and $M\mathbb{R}$ by

$$\forall A,B\in V\mathbb{R}:\quad A\leq B :\Longleftrightarrow A_i\leq B_i \quad \text{for } 1\leq i\leq n \text{ and}$$

$$\forall A,B\in M\mathbb{R}:\quad A\leq B :\Longleftrightarrow A_{ij}\leq B_{ij} \quad \text{for } 1\leq i,j\leq n.$$

The order relation $\leq$ in $\mathbb{C}$ is defined by

$$\forall\ a+bi,\ c+di\in\mathbb{C}:\quad a+bi\leq c+di :\Longleftrightarrow a\leq c \wedge b\leq d$$

and similarly extended to $V\mathbb{C}$ and $M\mathbb{C}$ (componentwise).

The sets $I\!I T$ of intervals over $\mathbb{R}, V\mathbb{R}, M\mathbb{R}, \mathbb{C}, V\mathbb{C}$ and $M\mathbb{C}$ are defined by

$$[A,B]\in I\!I T :\Longleftrightarrow [A,B] = \{x\in T \mid A\leq x\leq B\} \quad \text{for } A,B\in T.$$

Therefore $\mathbb{I}T \subseteq \mathbb{P}T$. We consider (see [23], [24]) a rounding $o: \mathbb{P}T \to \mathbb{I}T$ with the properties

(R) $\forall A \in \mathbb{P}T$: $oA = \cap\{B \in \mathbb{I}T \mid A \subseteq B\}$

(R1) $\forall A \in \mathbb{I}T$: $oA = A$

(R2) $\forall A, B \in \mathbb{P}T$: $A \subseteq B \Rightarrow oA \subseteq oB$

(R3) $\forall A \in \mathbb{P}T$: $A \subseteq oA$

(R4) $\forall o \neq A \in \mathbb{P}T$: $o(-A) = -(oA)$.

(R1), (R2) and (R3) are (together) equivalent to (R). Operations $\circledast$: $\mathbb{I}T \times \mathbb{I}T \to \mathbb{I}T$ for $* \in \{+, -, \cdot, /\}$ are defined by (see [21], [24])

(RG) $\forall A, B \in \mathbb{I}T$: $A \circledast B := o(A * B)$ $(= \cap\{C \in \mathbb{I}T \mid A * B \subseteq C\})$.

By means of semimorphisms it can be shown (cf. [21], [24]) that the operations in $\mathbb{I}T$ are well-defined (with well-known restrictions for /).

The operations are to be executed from left to right respecting the priorities and considering the canonical embeddings $T \subseteq \mathbb{I}T \subseteq \mathbb{P}T$ and $\mathbb{R} \subseteq \mathbb{C}$, $V\mathbb{R} \subseteq V\mathbb{C}$, $M\mathbb{R} \subseteq M\mathbb{C}$. To be perfectly clear we give the following example. Let $c \in \mathbb{C}$, $V \in \mathbb{I}V\mathbb{R}$, $A \in M\mathbb{R}$, $W \in \mathbb{I}V\mathbb{C}$. Then $c \odot V + A \odot W$ is well-defined. Following the rules of priorities first $X := c \cdot V$ is computed with $\cdot: \mathbb{C} \times \mathbb{I}V\mathbb{R} \to \mathbb{P}V\mathbb{C}$ and then rounded with o: $\mathbb{P}V\mathbb{C} \to \mathbb{I}V\mathbb{C}$. Then $Y := A \cdot W$ is computed with $\cdot: M\mathbb{R} \times \mathbb{I}V\mathbb{C} \to \mathbb{P}V\mathbb{C}$ and then rounded with o: $\mathbb{P}V\mathbb{C} \to \mathbb{I}V\mathbb{C}$. Finally $Z := oX + oY$ is computed with $+ : \mathbb{I}V\mathbb{C} \times \mathbb{I}V\mathbb{C} \to \mathbb{P}V\mathbb{C}$. It is well-known, that in fact $Z \in \mathbb{I}V\mathbb{C}$ (cf. [2], [24]). Moreover, in this specific case

$$c \cdot V + A \cdot W = c \odot V + A \cdot W \subseteq c \odot V + A \odot V = c \odot V \oplus A \odot W$$

For further details (cf. [21], [24]).

With S denoting a subset of $\mathbb{R}$ (e.g. the set of single-precision floating-point numbers on a computer) we consider the set $\mathbb{C}S$ of pairs over S, VS of n-tuples over S, MS of n^2-tuples over S, the set $V\mathbb{C}S$ of n-tuples over $\mathbb{C}S$ and $M\mathbb{C}S$ of n^2-tuples over $\mathbb{C}S$. Again, if the length of a vector or the number of rows of a quadratic matrix is other than n, resp. the matrix is not quadratic, the corresponding set has one, resp. two indices (e.g.

$V_{n+1}S$, $M_{\ell,m}\complement S$). Let U denote one of the sets $S, VS, MS, \complement S, V\complement S$ or $M\complement S$. Then intervals over one of these sets U are defined by

$$[A,B]\in \mathbb{I}U :\Longleftrightarrow [A,B] = \{x\in T \mid A\leq x\leq B\} \quad \text{for } A,B\in U,$$

where T is the set corresponding to U. The order relation $\leq$ is defined canonically by regarding U as a subset of T. We consider a rounding $\Diamond: \mathbb{I}T\to\mathbb{I}U$ with the properties (R), (R1), (R2) and (R3), (cf.,[22]). If U is symmetric ($U = -U$), then (R4) is also satisfied. The operations $\diamondsuit: \mathbb{I}U\times\mathbb{I}U\to\mathbb{I}U$ for $*\in\{+,-,\cdot,/\}$ are defined by (cf. [21], [24])

$$A \diamondsuit B := \Diamond (A \circledast B) \text{ for } A,B\in\mathbb{I}U. \tag{1.1}$$

It can be shown that

- $\Diamond$ is well-defined
- $\diamondsuit$ is well-defined for $*\in\{+,-,\cdot,/\}$
- $\diamondsuit$ is effectively implementable on a computer and
- $A \diamondsuit B = \cap\{C\in\mathbb{I}U \mid A\circledast B\subseteq C\}$ for $A,B\in\mathbb{I}U$.

These important properties are shown by means of algebraic and order isomorphism $\mathbb{I}V\mathbb{R}\longleftrightarrow V\mathbb{I}\mathbb{R}$, $\mathbb{I}M\complement S\longleftrightarrow M\mathbb{I}\complement S$ etc. (the operations in $V\mathbb{I}\mathbb{R}, M\mathbb{I}\complement S$ etc. are defined componentwise) and by explicitly giving algorithms for the operations $\diamondsuit$ in all sets S, VS, MS, $\mathbb{I}S$, $\mathbb{I}VS$, $\mathbb{I}MS$, $\complement S$, $V\complement S$, $M\complement S$, $\mathbb{I}\complement S$, $\mathbb{I}V\complement S$ and $\mathbb{I}M\complement S$ (cf. [21], [24]). For the latter purpose a precise arithmetic and Bohlender's algorithm (cf. [3]) are required. If T is the set corresponding to U then $\mathbb{I}U\subseteq\mathbb{I}T$ holds and therefore for $A,B\in\mathbb{I}U$ we have

$$\cap\{C\in\mathbb{I}U \mid A\circledast B\subseteq C\} = \cap\{C\in\mathbb{I}U \mid A*B\subseteq C\}.$$

Thus we extend the rounding $\Diamond$ to $\Diamond: \mathbb{P}T\to\mathbb{I}U$ by defining

$$A\in\mathbb{P}T: \Diamond(A) := \Diamond(oA).$$

If T is the set corresponding to U, then the monotone downwardly and monotone upwardly directed roundings $\nabla: T \to U$ and $\Delta: T \to U$ are defined by

$$A \in T: \diamondsuit([A,A]) = [\nabla A, \Delta A] \in I\!I\, U.$$

Similarly $\overline{\nabla}$ and $\overline{\Delta}$ are defined by (1.1). Finally we consider a rounding $\square: T \to U$ with the properties (R1), (R2) and (R3). For the rounding $\square$ and the corresponding operations $\boxed{*}: U \times U \to U$ defined by $a \boxed{*} b := \square(a * b)$ it can be shown (cf. [21], [24]) that

- $\square$ is well-defined
- $\boxed{*}$ is well-defined for $* \in \{+, -, \cdot, /\}$
- $\boxed{*}$ is effectively implementable on a computer and

for any $v \in U$ and $a,b \in U$ $a \boxed{*} b \leq v \leq a * b$ or $a * b \leq v \leq a \boxed{*} b$ implies $v = a \boxed{*} b$.

The final property holds for every set U and is called maximum accuracy. Let A,B be elements of $I\!P\, T$, $I\!I\, T$ or $I\!I\, U$. Then

$$A \underset{\neq}{\subset} B :\Longleftrightarrow \quad A \subseteq B \text{ and } A \neq B,$$

where the $\neq$ – sign is to be understood componentwise. $\mathring{A}$ denotes the topological interior of A and ∂A denotes the topological boundary of A. The absolute value of a vector, resp. matrix, over $\mathbb{R}$ or $\mathbb{C}$ is defined to be the vector, resp. matrix, of the absolute values of the components. For $A = [a,b] \in I\!I\, T$ with $a,b \in T$ the diameter $d(A) \in T$ and the absolute value $|A| \in T$ are defined by

$$d(A) := |b - a| \text{ and } |A| := \max(|a|, |b|),$$

where the maximum is to be understood componentwise.

For $X \in I\!I\, U$ with $X = [A,B]$; $A,B \in U$ we have

$$\inf(X) = A \text{ and } \sup(X) = B.$$

We define the "midpoint" of X by

$$m(X) := \inf(X) \boxplus (\sup(X) \boxminus \inf(X)) \boxslash 2 \in U. \tag{1.2}$$

It can be shown (cf. [33]) that with definition (1.2) $\inf(X) \leq m(X) \leq \sup(X)$ (as far as no over- or underflow occurs). In the following I denotes the $n \times n$ identity matrix. If the number of rows is other than n we write e.g. I_{n+1}. e_k denotes the k^{th} unit vector and $y_k := e_k' \cdot y$ the k^{th} component of a vector y. We consider a floating-point screen $S = S(b,\ell,e1,e2)$ where b is the base, ℓ is the length of the mantissa and $e1,e2$ are the smallest and largest possible exponent. In succeeding examples we refer e.g. to $S(10,12,-99,99)$ (this is the screen of our Z80-based minicomputer).

2. LINEAR SYSTEMS

Essential parts of this chapter have been introduced in [34]. Compared with the original work some theorems have been added and the proofs have been altered and/or simplified.

Let a system of linear equations $Ax = b$ with $A \in M\mathbb{R}$, $x,b \in V\mathbb{R}$ be given. For an approximate inverse $R \in M\mathbb{R}$ of A we have the residue iteration

$$x^{k+1} := x^k + R(b - Ax^k).$$

This iteration converges iff $\rho(I - RA) < 1$. If $X \in I\!P\,\mathbb{R}$ is some non-empty, convex, compact subset of $\mathbb{R}$ then by Brouwer's fixed point Theorem

$$X + R(b - AX) \subseteq X \text{ implies } \exists x \in X: R(b - Ax) = 0. \tag{2.1}$$

As a special non-empty, convex, compact subset of $\mathbb{R}$ we can choose $\emptyset \neq X \in I\!I\,V\mathbb{R}$. However, in general $d(X + R(b - AX)) > d(X)$ and $X + R(b - AX) \not\subseteq X$. Moreover, only if R is non-singular do we have an $x \in X$ with $Ax = b$. The two problems are solved by the next theorem (cf. [34]).

Theorem 2.1: Let $A,R \in M\mathbb{R}$ and $b \in V\mathbb{R}$. If then for some $X \in I\!I\,V\mathbb{R}$

$$Rb + \{I - RA\} \cdot X \subseteq \overset{\circ}{X}, \tag{2.2}$$

then R and A are non-singular and there is one and only one $\hat{x} \in \overset{\circ}{X}$ with $A\hat{x} = b$.

Proof: Define the function f: $V\mathbb{R} \to V\mathbb{R}$ by

$$f(x) := x + R(b - Ax) \tag{2.3}$$

and let $f(X) := \{f(x) \mid x \in X\}$. Then by (2.2) we have $f(X) \subseteq X$ and Brouwer's fixed point Theorem implies the existence of an $\hat{x} \in X$ with $f(\hat{x}) = \hat{x}$. Obviously $\hat{x} \in \overset{\circ}{X}$. By (2.3) we have $R(b - A\hat{x}) = 0$. For $y \in V\mathbb{R}$ with $Ay = 0$ and $\lambda \in \mathbb{R}$ brief computation shows

$$f(\hat{x} + \lambda y) = \hat{x} + \lambda y. \tag{2.4}$$

If $y \neq 0$, then a $\hat{\lambda}$ exists with $\hat{x} + \hat{\lambda} y \in \partial X$. This contradicts (2.4) and (2.2). Thus A is non-singular.

For $y \in V\mathbb{R}$ with $Ry = 0$ and $\lambda \in \mathbb{R}$ brief computation shows

$$f(\hat{x} + \lambda A^{-1} y) = \hat{x} + \lambda A^{-1} y. \tag{2.5}$$

If $y \neq 0$ then $A^{-1}y \neq 0$ and there exits a $\hat{\lambda} \in \mathbb{R}$ with $\hat{x} + \hat{\lambda} A^{-1} y \in \partial X$. This contradicts (2.5) and (2.2) hence R is non-singular. Therefore, $b - A\hat{x} \in Ker\, R = \{0\}$ and by the non-singularity of A the theorem is proved. □

In practical computation better error bounds are obtained when computing an inclusion of the difference of the exact solution and an approximate solution $\tilde{x}$ instead of computing an inclusion of the solution itself. This was observed in [34] and can be done using the following corollary.

Corollary 2.2: Let $A, R \in M\mathbb{R}$ and $\tilde{x}, b \in V\mathbb{R}$. If then for some $X \in \mathbb{I}\, V\mathbb{R}$

$$R(b - A\tilde{x}) + \{I - RA\} \cdot X \subseteq \overset{\circ}{X}, \tag{2.6}$$

then R and A are non-singular and there is one and only one $\hat{x} \in \tilde{x} + \overset{\circ}{X}$ with $A\hat{x} = b$.

Proof: Obvious. □

A direct consequence of theorem 2.1 and its proof is the extension to complex linear systems. We give a version for computing an inclusion of the difference between the exact solution and an approximate solution.

Theorem 2.3: Let $A,R \in M\mathbb{C}$ and $\tilde{x},b \in V\mathbb{C}$. If then for some $X \in \mathbb{I}V\mathbb{C}$

$$R(b - A\tilde{x}) + \{I - RA\} \cdot X \subseteq \overset{\circ}{X}, \tag{2.7}$$

then R and A are non-singular and there is one and only one $\hat{x} \in \tilde{x} + \overset{\circ}{X}$ with $A\hat{x} = b$.

There are no assumptions on $A,R,\tilde{x}$ or b required. The only provisions are (2.2), (2.6), (2.7).

Consider corollary 2.2 which gives a sufficient condition for existence and uniqueness of a solution of $Ax = b$ in $\tilde{x} + X$. If (2.6) does not hold one may initiate the following iteration process. Define f: $\mathbb{I}V\mathbb{R} \to \mathbb{P}V\mathbb{R}$ and F: $\mathbb{I}V\mathbb{R} \to \mathbb{I}V\mathbb{R}$ by

$$X \in \mathbb{I}V\mathbb{R}: f(X) := R(b - A\tilde{x}) + \{I - RA\} \cdot X, \quad F(X) := o(f(X)). \tag{2.8}$$

(In this particular case we have $F(X) = f(X)$). Let $F^0(X) := X$ and $F^{k+1}(X) := F(F^k(X))$ for $k \geq 0$. If then for some $X \in \mathbb{I}V\mathbb{R}$ and $k \in \mathbb{N}$ with $Y := F^k(X)$

$$f(Y) \subseteq \overset{\circ}{Y}$$

holds, then the assertions of Corollary 2.2 are valid. The question is, for which X, for which k and under which conditions this iteration will terminate. The answer is given by the following lemma.

Lemma 2.4: Let $|I - RA|$ be a primitive matrix and denote $\lambda := \rho(|I - RA|)$ and $z := R(b - A\tilde{x})$. Consider the mapping f defined by (2.8). Then the following are equivalent:

(A) For all $X \in \mathbb{IIVIR}$ with $|z + A \bullet m(X) - m(X)| < \frac{1-\lambda}{2} \bullet d(X)$ there exists a $k \in \mathbb{N}$ with $f(Y) \subseteq \overset{\circ}{Y}$, where $Y := F^k(X)$.

(B) $\rho(|I - RA|) < 1$.

Proof: cf. [36]. □

The final inclusion for the solution $\hat{x}$ of $Ax = b$ is $\tilde{x} + \overset{\circ}{X}$. So X may be chosen symmetric, i.e. $X = -X$. $|I - RA|$ is primitive if, for instance, $|I - RA|$ is positive. Moreover $m(X) = 0$ in this case, so the condition in A) of the preceding lemma reduces to

$$d(X) > \frac{2}{1-\lambda} \bullet |R(b - A\tilde{x})|. \tag{2.9}$$

Therefore, $\rho(|I - RA|) < 1$ if and only if the iteration terminates for every X satisfying (2.9). If $\tilde{x}$ is a good approximative solution of $Ax = b$ the absolute value of the components of $R(b - A\tilde{x})$ are small.

The essential advantage of corollary 2.2 is that it is applicable on computers.

Corollary 2.5: Let $A, R \in MS$ and $\tilde{x}, b \in VS$. If then for some $X \in \mathbb{IIVS}$

$$R \diamondsuit (b \diamondminus A \bullet \tilde{x}) \oplus (I \diamondminus R \bullet A) \diamondsuit X \subseteq \overset{\circ}{X}, \tag{2.10}$$

then R and A are non-singular and there is one and only one $\hat{x} \in \tilde{x} \oplus \overset{\circ}{X}$ with $A\hat{x} = b$.

Proof: Obvious by the definitions of $\diamondsuit$: $\mathbb{IIVIR} \to \mathbb{IIVS}$ and $\diamondsuit$: $\mathbb{IIVS} \times \mathbb{IIVS} \to \mathbb{IIVS}$ for $* \in \{+, -, \bullet, /\}$. □

According to [22], [24] and [3] (2.10) is executable on computers using the (effectively implementable) precise scalar product e.g. $I \diamondminus R \bullet A = \diamondsuit(o(I - RA)) = \diamondsuit(I - R \bullet A)$. The corollary remains true when replacing S by $\complement S$.

Now we are ready to give an algorithm for computing an inclusion of the solution of a system of linear equations which automatically verifies the correctness of the computed bounds.

1. Compute an approximate inverse R of A using your favorite algorithm;

2. $B := I \diamondminus R \cdot A$; $\tilde{x} := R \diamondcdot b$; $z := b \diamondminus A \cdot \tilde{x}$;, $z := R \diamondcdot z$; $X := z$;
   $k := 0$;

3. repeat $Y := X \diamondcdot [1-\varepsilon, 1+\varepsilon]$; $k := k+1$; $X := z \diamondplus B \diamondcdot Y$
   until $(X \subseteq \overset{\circ}{Y})$ or $(k = 10)$;

4. if $X \subseteq \overset{\circ}{Y}$ then {It has been verified, that the solution $\hat{x}$ of $Ax = b$
   exists and is uniquely determined and $\hat{x} \in \tilde{x} \diamondplus X$ holds}
   else {It could not be verified whether A is singular or not}.

Algorithm 2.1 Linear Systems

If the floating-point screen being employed is $S(b,\ell,e1,e2)$, then $\varepsilon := b^{-\ell+1}$ such that 1 and $1+\varepsilon$ are consecutive in S. This ε-inflation is introduced in [34], where its importance is demonstrated, too. The evaluations of B and z are executable on computers (cf. [4]). The assertion $\hat{x} \in \tilde{x} \diamondplus X$ instead of $\hat{x} \in \tilde{x} \diamondplus Y$ follows directly from (2.3) and (2.10). The including region $\tilde{x} \diamondplus X$ can be refined by

repeat $Y := X$; $X := (z \diamondplus B \diamondcdot Y) \cap Y$ until $X = Y$.

This iteration terminates because $d(X) \leq d(Y)$ in each step. Obviously every $\tilde{x} \diamondplus X$ includes $\hat{x}$.

In corollary 2.5 and algorithm 2.1, A and b are supposed to be elements of MS resp. VS. If this is not the case (for instance if A and b are the output of some measurement) consider the following theorem.

Theorem 2.6: Let $\mathcal{A} \in \mathbb{IM}\mathbb{R}$, $R \in M\mathbb{R}$, $\tilde{x} \in V\mathbb{R}$ and $b \in \mathbb{I}V\mathbb{R}$. If then for some $X \in \mathbb{I}V\mathbb{R}$

$$R(b - \mathcal{A}\tilde{x}) + \{I - R\mathcal{A}\} \cdot X \subseteq \overset{\circ}{X}, \tag{2.11}$$

then for every $A \in \mathcal{A}$ and every $b \in \mathbf{b}$ the following is true: A and R are non-singular and there is one and only one $\hat{x} \in \tilde{x} + \overset{\circ}{X}$ with $A\hat{x} = b$.

Proof: This follows because $R(b - A\tilde{x}) + \{I - RA\} \cdot X \subseteq R(\mathbf{b} - \mathcal{A}\tilde{x}) + \{I - R\mathcal{A}\} \cdot X$ for every $A \in \mathcal{A}$ and $b \in \mathbf{b}$ permitting the application of theorem 2.1. □

Again $\mathbb{IMR}$ and $\mathbb{IVR}$ may be replaced by $\mathbb{PMR}$, and $\mathbb{PVR}$ resp. if X is assumed to be non-empty, convex and compact. This remark is important when using a circular arithmetic in $\mathbb{C}$ (cf.[29]):

Corollary 2.7: Let $\mathcal{A} \in \mathbb{P}M\mathbb{C}$, $R \in M\mathbb{C}$, $x \in V\mathbb{C}$ and $\mathbf{b} \in \mathbb{P}V\mathbb{C}$. If then for some non-empty, convex and compact $X \in \mathbb{P}V\mathbb{C}$

$$R(\mathbf{b} - \mathcal{A}x) + \{I - R\mathcal{A}\} \cdot X \subseteq \overset{\circ}{X}, \tag{2.12}$$

then for every $A \in \mathcal{A}$ and every $b \in \mathbf{b}$ the following is true: A and R are non-singular and there is one and only one $\hat{x} \in \tilde{x} + \overset{\circ}{X}$ with $A\hat{x} = b$.

With theorem 2.6 and corollary 2.7 an algorithm can be derived for computing an inclusion of the solution of every linear system $Ax = b$ with $A \in \mathcal{A}$ and $b \in \mathbf{b}$ which automatically verifies the correctness of the computed bounds. This algorithm can be obtained by replacing A by $m(\mathcal{A})$ in step 1) and A by $\mathcal{A}$ resp. b by $\mathbf{b}$ in the computation of B and z in step 2) of algorithm 2.1. However, if some matrix A in $\mathcal{A}$ is ill-conditioned then $I \diamondsuit R\mathcal{A}$ may contain matrices of spectral radius 1.

For several improvements of the algorithms see [34].

Algorithm 2.1 can be used to verify the non-singularity of a matrix automatically on a computer. With $\tilde{x} := b := 0$ in corollary 2.2 we obtain

Corollary 2.8: Let $A, R \in M\mathbb{R}$. If then for some $X \in \mathbb{I}V\mathbb{R}$

$$(I - RA) \cdot X \subseteq \overset{\circ}{X}, \tag{2.13}$$

then A and R are not singular.

To verify whether or not a given matrix is singular is not a trivial problem. In [34] a 3×3 linear system is given which is exactly storable in the single-precision floating-point screen of the UNIVAC 1108 of the University of Karlsruhe. The system has been solved by Gaussian elimination with partial pivoting in single-precision accuracy (~8.5 decimal digits in the mantissa). Then a residue iteration was applied with double-precision evaluation of the residue (~19 decimal digits in the mantissa). The first and all iterates coincides with the initial approximation. Nevertheless, the matrix of the linear system is singular. If (2.13) does not hold for the initial X, one may use an iteration:

$$k:=0;\ \underline{\text{repeat}}\ k:=k+1;\ Y:=X;\ X:=(I-RA)\bullet Y\ \underline{\text{until}}\ X\subseteq\overset{\circ}{Y}; \tag{2.14}$$

The question for which X (2.14) terminates is answered by the following lemma:

Lemma 2.9: Let $A,R\in M\mathbb{R}$. If no entry of $I-RA$ is zero, then the following are equivalent:

a) For every initial $X\in\mathbb{I}V\mathbb{R}$ with $X=-X$ and $|X|>0$ the iteration (2.14) terminates.

b) $\rho(|I-RA|)<1$.

Proof: cf. [36]. □

The assumption $|I-RA|>0$ may be fulfilled by replacing a zero entry by b^{e1}, if $S(b,\ell,e1,e2)$ is the screen of the computer in use. The initial X may consist of $[-1,1]$ in every component according to the preceding lemma. The assertions of corollary 2.8 and lemma 2.9 remain valid when replacing $\mathbb{R}$ by $\mathbb{C}$. Moreover $(I-RA)\bullet X\subseteq(I\diamondsuit R\bullet A)\diamondsuit X$, so replacing (2.13) by $(I\diamondsuit R\bullet A)\diamondsuit X\subseteq\overset{\circ}{X}$ does not affect the assertion of corollary 2.8. So it is applicable on computers. Next we present an algorithm which verifies automatically the non-singularity of a matrix $A\in M\mathbb{R}$.

1. Compute an approximate inverse R of A using your favorite algorithm;

2. $B := I \diamondsuit R \bullet A$; $X := ([-1,1])$; $k := 0$;

3. repeat $k := k + 1$; $Y := X \diamondsuit (1 + \varepsilon)$; $X := B \diamondsuit Y$ until $(X \subseteq \overset{\circ}{Y})$ or $(k = 10)$;

4. if $X \subseteq \overset{\circ}{Y}$ then {It has been verified A is not singular}
   else {It could not be verified, whether A is singular or not}.

Algorithm 2.2 Non-singularity of a matrix

It should be mentioned, that it is not possible to verify the singularity of a matrix without computing exactly in the field of reals (for instance using an exact integer or rational number package). This is because in any ε-neighborhood of a singular matrix there are regular matrices. Algorithm 2.2 is directly applicable to complex matrices $A \in M\mathbb{C}$ when one replaces the initial X by $([-1-i, 1+i])$. After replacing A by $\mathcal{A} \in \mathbb{I}M\mathbb{R}$ or $\mathcal{A} \in \mathbb{I}M\mathbb{C}$ it can be determined, whether every matrix $A \in \mathcal{A}$ is non-singular.

If the matrix of the linear system is symmetric, the presented algorithms can easily be improved resulting in a computing time of $\sim n^3$, i.e., one half the computing time of systems with general matrix.

Algorithm 2.2 may be used to verify, that a symmetric matrix $A \in M\mathbb{R}$ is positive definite or, that a matrix $A \in M\mathbb{R}$ or $\mathcal{A} \in \mathbb{I}M\mathbb{C}$ has only eigenvalues with positive real part. If $D := I - \|A\|^{-1} \bullet A$ for some norm $\|\bullet\|$, then $\rho(D) < 1$ implies

a) $Ax = (\lambda + i\mu)x \Rightarrow \lambda > 0$ and

b) A symmetric $\Rightarrow$ A is positive definite.

There are similar applications to $\mathcal{A} \in M\mathbb{R}$ or $M\mathbb{C}$ in case a). Bounds for the real resp. imaginary parts of the eigenvalues of $A \in M\mathbb{R}$ may be obtained by estimating the eigenvalues of the symmetric resp. antisymmetric part $\frac{1}{2}(A + A^T)$ resp. $\frac{1}{2}(A - A^T)$.

All of the preceding theorems and corollaries in this chapter remain true when replacing $\subseteq^{\circ}X$ by $\underset{\neq}{\subset}X$. For a proof of this fact cf. [36].

The computing time of algorithm 2.1 when using the Gauss-Jordan algorithm in step 1 is $2n^3 + 3n^2 + 2kn^2$. If the matrix of the linear system has special properties like symmetry or positive definiteness the computing time of the presented algorithms can be reduced significantly. In general with $k \leq 10$ the ratio of computing times between algorithm 2.1 and Gaussian elimination is $\leq 6 + 0\left(\frac{1}{n}\right)$ (cf. [34]). However, every result given by any algorithm discussed in this chapter is verified to be correct.

Next we discuss some computational results. As a small ill-conditioned example consider

$$\begin{pmatrix} 37639840 & -46099201 \\ 29180474 & -35738642 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

For this system the standard built-in algorithm of a very common computer with mantissa length 16 decimal places computes the approximation

$$\begin{pmatrix} 28869851.52297299 \\ 23572135.06039856 \end{pmatrix},$$

whereas on our computer with mantissa length 12 decimal digits the following inclusion were computed:

$$\begin{pmatrix} [46099201.0,\ 46099201.0] \\ [37639840.0,\ 37639840.0] \end{pmatrix}.$$

The inclusion is a point and therefore exact. The correctness is verified by the computer automatically.

We define the

Hilbert matrix H^n by $H^n_{ij} := 1cm(1,2,\ldots,2n-1)/(i+j-1)$,

Pascal matrix P^n by $P^n_{ij} := \binom{i+j}{i}$

Pascal* matrix P^{*n} by $P^{*n}_{ij} := \binom{i+j-2}{i-1}$

the matrix Q^n by $Q^n_{ij} := \dfrac{\binom{n+j-1}{i-1} \cdot n \cdot \binom{n-1}{j-1}}{i+j-1}$.

The matrices with maximum number of rows exactly storable in the screen $S(10,12,-99,99)$ of our minicomputer are H^{15}, P^{21}, P^{*22}, Q^{16}. Here are the computational results for these matrices with right hand side (1,...,1):

> Every linear system with the matrices H^n, P^n, P^{*n} and Q^n up to the maximum number of rows for which the matrices are exactly storable in $S(10,12,-99,99)$ has been solved with automatic verification of the non-singularity of the matrix and with least significant bit accuracy in every component of the solution. The condition number of the Hilbert 15×15 matrix is approximately 10^{22}, the condition number of Q^{16} is $2\cdot10^{24}$.

The approximations for the components of the solution computed by a (purely) floating-point algorithm may be arbitrarily false. Using Gaussian elimination, for instance, yields for one component of the solution of the linear system with Hilbert 15×15 matrix and right hand side (1,1,...,1) an approximation

3471.76599106

whereas the inclusion computed by the new methods is

0.00099900099900^{1}_{0}.

Least significant bit accuracy for an inclusion of a component means that the left and right bound of the inclusion are consecutive numbers in the floating-point screen. Linear systems with dense matrix and up to 210 rows have been treated on the UNIVAC 1108 of the University of Karlsruhe. Here the size is only limited by the memory of the machine. In every case the least significant bit accuracy property holds for every component of the solution.

As a final example take a linear system with Hilbert 21×21 matrix with right hand side (1,0,0,...,0). As in the previous example the components of the matrix are multiplied by a proper factor to obtain integer entries. The computer in use is a IBM 370/168, double precision (i.e. 14 hexadecimal digits in the mantissa). The Hilbert 21×21 matrix is the matrix of largest number of rows exactly storable in that floating-point screen. Here are the results of ordinary Gaussian elimination compared with the new algorithm:

Gaussian elimination	new algorithm
$0.7176601221737417D-15$	$0.201314533929{80}_{29}^{30}D-14$
$-0.5463879586639182D-13$	$-0.442891974645{566}_{5}^{4}D-12$
$0.1300043029451921D-11$	$0.3225729882001{87}_{5}^{6}D-10$
$-0.1384647901664727D-10$	$-0.11612627575206{7}_{6}^{5}D-08$
$0.7307664917097330D-10$	$0.24676833597314{3}_{4}^{5}D-07$
$-0.1742770450134503D-09$	$-0.342185425882{75}_{90}^{89}D-06$
$0.4670356473353298D-10$	$0.32996451781551{7}_{5}^{6}D-05$
$0.2755923072731661D-09$	$-0.23097516247086{2}_{3}^{2}D-04$
$0.1934981248757405D-08$	$0.12094116146043{7}_{6}^{7}D-03$
$-0.1055453079740496D-07$	$-0.48376464584175{0}_{5}^{4}D-03$
$0.2015964106800396D-07$	$0.14996704021094{2}_{6}^{7}D-02$
$-0.1947461412799036D-07$	$-0.36355646111743{6}_{8}^{7}D-02$
$0.1797737164979233D-07$	$0.69215557020435{0}_{7}^{8}D-02$
$-0.3066524812704846D-07$	$-0.10344303027229{8}_{6}^{5}D-01$
$0.2853218866898916D-07$	$0.12068353531768{1}_{6}^{7}D-01$
$-0.1245274389638609D-07$	$-0.10861518178591{3}_{5}^{4}D-01$
$-0.3920468862403904D-07$	$0.73874296435272{0}_{4}^{5}D-02$
$0.1162363936934045D-07$	$-0.36695728948239{7}_{1}^{0}D-02$
$0.2062283997953500D-07$	$0.12553802008608{3}_{2}^{3}D-02$
$-0.1794039988466323D-07$	$-0.26429056860228{0}_{5}^{4}D-03$
$0.4327580718388825D-08$	$0.25799793601651{1}_{8}^{9}D-04$

3. OVER- AND UNDERDETERMINED LINEAR SYSTEMS

Let $A \in M_{\ell,m}\mathbb{R}$ and $b \in V_\ell\mathbb{R}$. For $\ell > m$, the linear system is overdetermined with, in gneral, no solution and the vector $\hat{x} \in V_m\mathbb{R}$ is to compute such that $\|b - A\hat{x}\|$ is minimal.†

If $\ell < m$ we have an underdetermined system. In general, there are infinitely many solutions and the vector $\hat{y} \in V_m\mathbb{R}$ is to compute for which $A\hat{y} = b$ and $\|\hat{y}\|$ is minimal. If the rank of A is maximal, the solution for both problems is uniquely determined. It is well-known (cf. [39]), that if

$$\ell > m \text{ and } rg(A) = m \text{ then } \hat{x} \text{ is the solution of } A^TAx = A^Tb \tag{3.1}$$

$$\ell < m \text{ and } rg(A) = \ell \text{ then } \hat{y} = A^Tx, \text{ where } AA^Tx = b. \tag{3.2}$$

The linear systems occurring in (3.1) and (3.2) may be solved with algorithm 2.1 of the preceding chapter. However, in general A^TA and AA^T are real matrices and not elements of MS. Thus the interval version using some $\mathcal{A} \supseteq A^TA$ resp. $A \supseteq AA^T$ with $\mathcal{A} \in \mathbb{I}MS$ would have to be used. This is out of the question because A^TA and AA^T are in general ill-conditioned. Instead we use the following linear systems:

$$\begin{pmatrix} A & -E \\ 0 & A^T \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} \quad \text{for } \ell > m, \tag{3.3}$$

$$\begin{pmatrix} A^T & -E \\ 0 & A \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix} \quad \text{for } \ell < m. \tag{3.4}$$

Then a short computation demonstrates the following theorems.

† In this chapter $\|\cdot\|$ denotes the Euclidean norm.

Theorem 3.1: Let $A \in M_{\ell,m}\mathbb{R}$, $b \in V_\ell\mathbb{R}$, $\ell > m$ and define $C \in M_{\ell+m,\ell+m}\mathbb{R}$ to be the square matrix in (3.3). Define $b^* \in V_{\ell+m}\mathbb{R}$ to be the vector $(b, 0)$†† and let $\tilde{z} \in V_{\ell+m}\mathbb{R}$, $R \in M_{\ell+m,\ell+m}\mathbb{R}$. If then for some $Z \in \mathbb{I}V_{\ell+m}\mathbb{R}$

$$R(b^* - C\tilde{z}) + \{I - RC\} \cdot Z \subseteq \overset{\circ}{Z}, \tag{3.5}$$

then there is an $\hat{x} \in \tilde{x} + X$ with the following property:

For any $x \in V_m\mathbb{R}$ with $x \neq \hat{x}$ holds $\|b - A\hat{x}\| < \|b - Ax\|$,

where x resp. X are the first m components of $\tilde{z}$ resp. Z and I is the $(\ell + m) \times (\ell + m)$ unit matrix. Further the matrix A has maximum rank m.

Theorem 3.2: Let $A \in M_{\ell,m}\mathbb{R}$, $b \in V_\ell\mathbb{R}$, $\ell < m$ and define $C \in M_{\ell+m,\ell+m}\mathbb{R}$ to be the square matrix in (3.4). Define $b^* \in V_{\ell+m}\mathbb{R}$ to be the vector $(0,b)$ and let $\tilde{z} \in V_{\ell+m}\mathbb{R}$, $R \in M_{\ell+m,\ell+m}\mathbb{R}$. If then for some $Z \in \mathbb{I}V_{\ell+m}\mathbb{R}$

$$R(b^* - C\tilde{z}) + \{I - RC\} \cdot Z \subseteq \overset{\circ}{Z}, \tag{3.6}$$

then there is an $\hat{y} \in \tilde{y} + Y$ with the following properties:

a) $A\hat{y} = b$

b) if $Ay = b$ for some $y \in V_m\mathbb{R}$ with $y \neq \hat{y}$ then $\|\hat{y}\| < \|y\|$,

where $\tilde{y}$ resp. Y are the last m components of $\tilde{z}$ resp. Z and I is the $(\ell + m) \times (\ell + m)$ unit matrix. Further the matrix A has maximum rank ℓ.

Both theorems are applicable on computers. Here one replaces $\mathbb{R}$ by S, the conditions (3.5) and (3.6) by $R \diamondsuit (b^* \diamondsuit C \cdot \tilde{z}) \diamondsuit \{I \diamondsuit R \cdot C\} \diamondsuit Z \subseteq \overset{\circ}{Z}$, resp. and $\tilde{x} + X$ resp. $\tilde{y} + Y$ by $\tilde{x} \diamondsuit X$ resp. $\tilde{y} \diamondsuit Y$. Here, for instance, $b^* \diamondsuit C \cdot \tilde{z} = \diamondsuit(o(b^* - C\tilde{z}))$ is effectively computable (cf. [21], [24], [3]). However, the computing time for these algorithms is $2(\ell + m)^3$ compared with pq^2 for the orthogonalization method (cf. [39]) where

†† $(b, 0)$ is the vector in $V_{\ell+m}\mathbb{R}$ such that the first ℓ components are those of b and the remaining m components are zero. We use similar notations frequently.

$p = \max(\ell,m)$, $q = \min(\ell,m)$. Utilizing symmetries and the fact that not every component of R and $I - RC$ has to be computed explicitly (this is an extended escalator method) results in a

$$\text{computing time } 3pq^2 + 2p^2q + q^3/3$$

(cf. [7]). Therefore the time for computing guaranteed bounds for the solution with automatic verification of both correctness and maximum rank of A is for $p = 2q$ approximately seven times the computing time for a usual floating-point computation. As is well-known, least square approximation problems are in general ill-conditioned. Therefore guaranteed bounds gain in significance. As well as for the least square approximating polynomial bounds for the coefficients of the interpolation polynomial can be included. In the latter case the computing time can be reduced from $16n^3$ to $5n^3$ by utilizing the special structure of the matrix.

As in the case of linear systems with square matrices the algorithm can be extended to matrices $\mathscr{A} \in \mathbb{IM}_{\ell,m}S$ and vectors $\mathscr{b} \in \mathbb{IV}_\ell S$. In this case with corresponding conditions (3.5) and (3.6) every least square problem $Ax = b$ with $A \in \mathscr{A}$ and $b \in \mathscr{b}$ is solvable, every matrix $A \in \mathscr{A}$ has maximum rank and the uniquely determined solution is included in the corresponding interval $\tilde{x} \diamondsuit X$ resp. $\tilde{y} \diamondsuit Y$. Once again automatic verification of correctness is accomplished by the algorithm without any additional effort on the part of the user.

As in the case of linear systems with equal number of rows and unknowns there are extensions in the field of complex numbers in the cases of underdetermined and overdetermined linear systems. The corresponding theorems can be extended to $\mathbb{C}S$ in the same manner as in chapter 2.

The methods and algorithms described in this chapter can be used to include the pseudoinverse of a matrix with automatical verification of correctness.

Numerical examples for least square approximation are taken from [39], Beispiel 3.6. The problem is to find a polynomial P of degree n through $N + 1$ given points (x_i,y_i), $i = 1(1)N + 1$ which minimizes $\Sigma(P(x_i) - y_i)^2$. We choose the abzissas $x_i = i$. The

elements A_{ij} of the matrix A are given by $A_{ij} = i^{j-1}$, $i = 1(1)N + 1$, $j = 1(1)n + 1$. According to (3.19) in [39] the ratio κ between the largest and smallest eigenvalue of A^TA can be estimated by

$$\kappa \geq \max_k A_{kk} / \min_k A_{kk}.$$

Therefore, in our particular case

$$\kappa \geq \left(\sum_{i=1}^{N+1} i^{2n} \right) / (N + 1).$$

In the following we give a table for the minimal value of κ for different n,N.

n \ N	10	11	12	13	14	15
10	$7.1_{10}19$	$3.8_{10}20$	$1.8_{10}21$	$7.7_{10}21$	$2.9_{10}22$	*
11		$5.4_{10}22$	*	*	*	*

The maximal component of A is $(N + 1)^n$. An asterisk in the table above indicates that some components of the specific matrix A are not exactly storable in our computer with the floating-point screen $S(10,12, -99,99)$. Computational results:

> Every least-square problem for the values of n,N without entry * in the table above has been solved by the new algorithm to least significant bit accuracy. This means that the left and right bound of every component are consecutive numbers in the floating-point screen of the computer. The correctness of every result is verified by the computer automatically.

As seen from the table the examples are ill-conditioned, but nevertheless solved with automatic error control on a computer with a 12 decimal digit floating-point screen.

As a small ill-conditioned example consider

$$\begin{pmatrix} 665857 & -941664 \\ 470832 & -665857 \\ 470833 & -665857 \end{pmatrix} \bullet \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 665858 \end{pmatrix}$$

The computed inclusion of the vector with smallest sum of the norms of the residues is

$$\begin{pmatrix} [665858.000000,\ 665858.000001] \\ [470832.707107,\ 470832.707108] \end{pmatrix},$$

whereas the floating-point approximation is

$$\begin{pmatrix} 665700.0 \\ 470900.0 \end{pmatrix}.$$

4. LINEAR SYSTEMS WITH BAND MATRICES

The inverse of a band matrix is, in general, dense. Thus the algorithms presented in chapter 2 are too time consuming in this case . There is another possibility for computing an approximate inverse $R \in M\mathbb{R}$ of a matrix $A \in M\mathbb{R}$. Instead of R itself a LU-decomposition of A is computed with lower and upper triangular matrices $L,U \in M\mathbb{R}$. Then $R = (LU)^{-1} = U^{-1}L^{-1} = A^{-1}$. Of course, neither L,U nor R is determined exactly by the computer; they are merely approximated.

All theorems and corollaries of chapter 2 remain valid when one replaces R by $U^{-1} \bullet L^{-1}$. Let us consider the typical condition

$$R(b - A\tilde{x}) + \{I - RA\} \bullet X \subseteq \overset{\circ}{X}. \tag{4.1}$$

$L^{-1} \bullet c$ and $U^{-1} \bullet c$ for $c \in V\mathbb{R}$ can always be computed by backward substitution provided that the diagonal elements of L and U are not zero. We denote this process by $L^{-1} * c$ resp. $U^{-1} * c$. Thus (4.1) may be replaced by

$$U^{-1} * (L^{-1} * (b - A\tilde{x})) + \{I - U^{-1} * (L^{-1} * A)\} \bullet X \subseteq \overset{\circ}{X}, \tag{4.2}$$

where in $L^{-1}*A$ and $U^{-1}*(L^{-1}*A)$ backward substitution is applied columnwise.

Next we describe backward substitution over S with the rounding $\diamondsuit$. Suppose $L = (L_{ij})$ with $L_{ij} = 0$ for $i<j$, $L_{ii}\neq 0$. Then for $c\in V\mathbb{R}$

$$L^{-1}*c = v \text{ with } v = (v_i)\in V\mathbb{R} \text{ and } v_i = (c_i - \sum_{j=1}^{i-1} L_{ij}v_j)/L_{ii}. \tag{4.3}$$

v_i is computed for $i = 1(1)n$. If $L\in MS$ and $c\in VS$, then

$$L^{-1} \diamondsuit^* c := v \text{ with } v = (v_i)\in \mathbb{I}VS \text{ and } v_i := (c_i \diamondminus \sum_{j=1}^{i-1} L_{ij}v_j) \diamondsuit L_{ii}. \tag{4.4}$$

The definition for $U^{-1} \diamondsuit^* c$ is similar. The operations in (4.4) are executable on computers (cf. [21], [24], [3]). It is easy to see that

$$L^{-1}*c \subseteq L^{-1} \diamondsuit^* c. \tag{4.5}$$

For $A\in MS$ we define $L^{-1} \diamondsuit^* A$ to be the matrix consisting of columns $L^{-1} \diamondsuit^* A_i$, where A_i are the columns of A.

Therefore the theorems and corollaries of chapter 2 can be reformulated using a LU-decomposition instead of R. As an example we give such a version for corollary 2.5.

Theorem 4.1: Let $A,L,U\in MS$ and $\tilde{x},b\in VS$ where L resp. U are lower resp. upper triangular with non-zero diagonal elements. If then for some $X\in \mathbb{I}VS$

$$U^{-1} \diamondsuit^* (L^{-1} \diamondsuit^* (b \diamondminus A\cdot\tilde{x})) \diamondplus \{I - U^{-1} \diamondsuit^* (L^{-1} \diamondsuit^* A)\} \diamondsuit X \subseteq \mathring{X}, \tag{4.6}$$

then L,U and A are non-singular and there is one and only one $\hat{x}\in\tilde{x} + X$ with $A\hat{x} = b$.

Remark: The subtraction of $U^{-1} \diamondsuit^* (L^{-1} \diamondsuit^* A)$ from the unit matrix is integrated in $U^{-1} \diamondsuit^* B$, where $B := L^{-1} \diamondsuit^* A$. Therefore the minus-sign in the braces is written without a rounding.

For applications to band matrices theorem 4.1 has the advantage, that the band structure is not destroyed. Therefore the computing time is significantly reduced. If for the band matrix A having $A_{ij} = 0$ for $|i - j|>m$ then computation of a LU-decomposition of A

costs about nm^2. Therefore the total computing time for (4.6) is $\sim 4nm^2$. If the matrix has special properties like symmetry or positive definiteness the time can again be reduced significantly. In general there is a factor 4 in cost compared with a usual floating-point algorithm. In return one gains the automatic verification of the non-singularity of the given matrix. Therefore the solvability of the system is demonstrated by the algorithm without any effort on the part of the user.

Similar to chapter 2 the presented methods are applicable in the field of complex numbers as well as for $\mathcal{A} \in \mathbb{IIMS}$, $b \in \mathbb{IIVS}$ resp. $\mathcal{A} \in \mathbb{IIM}\mathbb{C}S$, $b \in \mathbb{IIV}\mathbb{C}S$. In the two latter cases the non-singularity of every $A \in \mathcal{A}$ is automatically verified, and in this case for every $A \in \mathcal{A}$, $b \in \mathcal{b}$ there is an $\hat{x} \in \tilde{x} \diamondsuit X$ with $A\hat{x} = b$.

A disadvantage of (4.6) is that $U^{-1} \diamondsuit L^{-1} \diamondsuit c$ and $I - U^{-1} \diamondsuit (L^{-1} \diamondsuit A)$ with $c := b \diamondsuit A\tilde{x}$ is computed with more than one rounding. Therefore extremely ill-conditioned systems are not solvable. The author's experience showed that in a floating-point screen $S(10,12,-99,99)$ linear systems with matrices up to condition numbers $5 \cdot 10^8$ are solvable with least significant bit accuracy for each component of the solution.

5. SPARSE LINEAR SYSTEMS

Let $A \in M\mathbb{R}$, $b \in V\mathbb{R}$. We consider an iteration scheme (cf. [41])

$$x^{i+1} := x^i + B^{-1} \cdot (b - Ax^i) \tag{5.1}$$

for some initial $x^0 \in V\mathbb{R}$ and an iteration matrix $B \in M\mathbb{R}$. (5.1) converges iff $\rho(I - B^{-1}A) < 1$. Let $A := L + D + U$, where $L, U, D \in M\mathbb{R}$ are lower, upper and diagonal matrices. Then we get for $B := D$ the

$$\text{Jacobi method:} \qquad x^{i+1} := x^i + D^{-1} \cdot (b - Ax^i), \tag{5.2}$$

for $B := D + L$ the

$$\text{Gauss} - \text{Seidel method:} \qquad x^{i+1} := x^i + (D + L)^{-1} \cdot (b - Ax^i) \tag{5.3}$$

and for $B := \omega^{-1}(D + \omega L)^{-1}$, $\omega \in \mathbb{R}$ the

relaxation method: $$x^{i+1} := x^i + \omega \bullet (D + \omega L)^{-1} \bullet (b - Ax^i). \tag{5.4}$$

The methods are well-defined if the diagonal elements of A do not vanish. We distinguish the three methods by

$$B_1 := D;\ B_2 := D + L;\ B_3 := \omega^{-1}(D + \omega L) \text{ for fixed } 0 \neq \omega \in \mathbb{R}. \tag{5.5}$$

Each of the three methods can be used to compute inclusions of the solution of a linear system as one sees by replacing R by B_1, B_2 or B_3 in Theorem 2.1. However, the computing time would be approximately $n^3/3$ which is out of the question. Next we present methods based on (5.2), (5.3), (5.4), resp. for computing bounds for the solution of a linear system in computing time $n^2/2$ per step.

Theorem 5.1: Let $\mathcal{A} \in M\mathbb{R}$, $b \in V\mathbb{R}$ with $A = L + D + U$ for lower, upper and diagonal matrices $L, U, D \in M\mathbb{R}$. Suppose the diagonal elements of A do not vanish. If then for some $X \in \mathbb{IV}\mathbb{R}$ one of the following conditions is satisfied:

1) $D^{-1} \bullet \{b - A\tilde{x} - (L + U)X\} \subseteq \overset{\circ}{X}$

2) $(D + L)^{-1} \bullet \{b - A\tilde{x} - UX\} \subseteq \overset{\circ}{X}$

3) $(\omega^{-1}D + L)^{-1} \bullet \{b - A\tilde{x} - (U + D - \omega^{-1}D)X\} \subseteq \overset{\circ}{X}$ for some $0 \neq \omega \in \mathbb{R}$,

then the matrix A is not singular and there exists an $\hat{x} \in \tilde{x} + X$ such that $A\hat{x} = b$.

Proof: Define the three functions f_i: $V\mathbb{R} \to V\mathbb{R}$, $i = 1(1)3$ by

$$x \in V\mathbb{R}: f_i(x) := B_i^{-1} \bullet b + \{I - B_i^{-1} \bullet A\} \bullet x \tag{5.6}$$

for fixed $0 \neq \omega \in \mathbb{R}$ in case $i = 3$.

($B_i, i = 1(1)3$ is given in (5.5)). Then a brief computation shows that if assumption i) is satisfied, $i = 1(1)3$ then with $Y := \tilde{x} + X$, $Y \in \mathbb{IV}\mathbb{R}$ we have

$$f_i(Y) \subseteq \overset{\circ}{Y}.$$

Whence applying theorem 2.1 we have the non-singularity of A and the assertion $\hat{x} \in Y = \tilde{x} + X$ with $A\hat{x} = b$ if 1), 2) or 3) is satisfied. □

The preceding theorem is applicable on computers as is shown in the following corollary.

Corollary 5.2: Let $A \in MS$, $b \in MS$ with $A = L + D + U$ for lower, upper and diagonal matrices $L,U,D \in MS$. Suppose the elements of the diagonal of A do not vanish. If then for some $X \in \mathbb{IV}S$ one of the following conditions is satisfied:

A) $D^{-1} \diamondsuit \Diamond \{b - A \cdot \tilde{x} - (L + U) \cdot X\} \subseteq \mathring{X}$

B) $(D + L)^{-1} \diamondsuit \Diamond \{b - A \cdot \tilde{x} - U \cdot X\} \subseteq \mathring{X}$

C) $(\omega^{-1}D + L)^{-1} \diamondsuit \Diamond \{b - A\tilde{x} - (U + D - \omega^{-1}D) \cdot X\} \subseteq \mathring{X}$

for fixed $0 \neq \omega \in \mathbb{R}$ in case $i = 3$,

then the matrix A is non-singular and there exists an $\hat{x} \in \tilde{x} \diamondsuit X$ with $A\hat{x} = b$.

Here $\Diamond \{b - A\tilde{x} - (L + U) \cdot X\} \in \mathbb{IV}S$, $\Diamond \{b - A\tilde{x} - U \cdot X\} \in \mathbb{IV}S$ and $\Diamond \{b - A\tilde{x} - (U + D - \omega^{-1}D) \cdot X\} \in \mathbb{IV}S$ are effectively computable using one of the algorithms in [3] (cf. [4], too). The symbol $\diamondsuit$ is defined in the previous chapter.

As demonstrated in chapter 2 the above methods are applicable in the field of complex numbers as well as for $\mathcal{A} \in \mathbb{IM}S$, $\mathcal{b} \in \mathbb{IV}S$ resp. $\mathcal{A} \in \mathbb{IM}\mathbb{C}S$, $\mathcal{b} \in \mathbb{IV}\mathbb{C}S$. In the two latter cases the non-singularity of every $A \in \mathcal{A}$ is verified automatically and for every $A \in \mathcal{A}$, $b \in \mathcal{b}$ there is an $\hat{x} \in \tilde{x} \diamondsuit X$ with $A\hat{x} = b$.

Research on sparse linear systems is in progress. Up to now we have little experience on the range applicability of the algorithms. As a numerical example consider linear system (8.4.5) on page 246, [41]. The matrix derives from a discretization of the Dirichlet boundary value problem $-u_{xx} - u_{yy} = f(x,y)$ for $0 < x,y < 1$ and $u(x,y) = 0$ for $(x,y) \in \partial\Omega$ with $\Omega := \{(x,y) \mid 0 < x,y < 1\}$. Let $N = 32$ which corresponds to a linear system with 1024 unknowns approximately 11 steps were needed using (5.4) with optimal relaxation factor to reduce the relative error of an approximate solution to 1/10 (this corresponds to (8.4.9) in [41]). Using the extended relaxation method C) in corollary 5.2 with optimal relaxation factor we achieved least significant bit accuracy for every component of the solution with

automatic verification of correctness. Of course, the iteration could be terminated earlier if this high accuracy is not required.

6. MATRIX INVERSION

For $A \in M\mathbb{R}$ we consider the problem of finding a matrix $\hat{C}$ with $A \cdot \hat{C} = I$. Given an initial approximation $R^0 \in M\mathbb{R}$, a Newton-like iteration can be carried out (Schulz-procedure, cf. [2]):

$$R^{i+1} := R^i + R^i(I - AR^i), \quad i \geq 0. \tag{6.1}$$

By methods analogous to those derived in chapter 2 an inclusion of the inverse $\hat{C}$ can be computed.

Theorem 6.1: Let $A, R \in M\mathbb{R}$. If then for some $X \in \mathbb{I}M\mathbb{R}$

$$R(I - A \cdot R) + \{I - R \cdot A\} \cdot X \subseteq \overset{\circ}{X}, \tag{6.2}$$

then the matrices A and R are non-singular and there is a $\hat{C} \in R + \overset{\circ}{X}$ with $A \cdot \hat{C} = I$.

Proof: The non-singularity of A and R follows as in the proof of theorem 2.1. Further for $f: M\mathbb{R} \to M\mathbb{R}$ with $f(D) := D + R(I - AD)$, $D \in M\mathbb{R}$ we conclude after a brief computation (6.2) that

$$f(Y) \subseteq \overset{\circ}{Y} \quad \text{where} \quad Y := R + X.$$

Therefore by Brouwer's fixed point theorem there is a $\hat{C} \in Y = R + \overset{\circ}{X}$ with $f(\hat{C}) = \hat{C}$. This implies $R(I - A\hat{C}) = 0$ and by the non-singularity of R we have $A\hat{C} = I$. □

The preceding theorem can be extended to all of the cases A in MS, $\mathbb{I}MS$, $M\mathcal{C}$, $M\mathcal{C}S$ and $\mathbb{I}M\mathcal{C}S$ in a manner similar to that demonstrated in chapter 2.

An inclusion of $\hat{C}$ can be obtained columnwise by applying the theorems and corollaries of chapter 2 to the linear systems $AX = e_i$, where e_i are the columns of I. The computing time for both processes is the same namely $\sim 4n^3$. The iteration (6.1) has the advantage, that in every iteration step a new improved inverse is used. This is not the case when using (2.1).

If, on the other hand, only a few elements of the inverse are to be computed, the second method is preferable, because it is faster and needs less memory.

As an example consider

$$A := \begin{pmatrix} 941664 & 665857 \\ 665857 & 470832 \end{pmatrix}.$$

Inverting A using the Gauss-Jordan procedure in $S(10,12,-99,99)$ yields an approximate inverse

$$R = \begin{pmatrix} -166666.666667 & 235702.260396 \\ 235702.260396 & -333333.333333 \end{pmatrix}.$$

The new algorithm computes an inclusion of A^{-1} to least significant bit accuracy with automatic verification of both correctness and non-singularity of A:

$$A^{-1} \in \begin{pmatrix} [-470832.0,-470832.0] & [665857.0,665857.0] \\ [665857.0,665857.0] & [-941664.0,-941664.0] \end{pmatrix}.$$

In this special case the resulting left and right bounds coincide, i.e. the inclusion of A^{-1} is a point matrix. Consider

$$A := \begin{pmatrix} [(941664,941664.000001] & 665857 \\ 665857 & 470832 \end{pmatrix} \in \mathbb{IMS}.$$

In this case only the first component of A has been replaced by to an interval of smallest possible diameter in the screen $S(10,12,-99,99)$. The computed inclusion is

$$A \in \mathscr{A} \Rightarrow A^{-1} \in \begin{pmatrix} [-8.9_{10}5,-4.7_{10}5] & [1.2_{10}5,6.6_{10}5] \\ [1.2_{10}5,6.6_{10}5] & [-9.5_{10}5,-1.7_{10}5] \end{pmatrix}.$$

These bounds are as small as possible and display the fact that A is ill-conditioned.

For every matrix H^n, P^n,P^{*n} and Q^n defined in chapter 2 up to the highest number of rows for which the matrix is exactly storable in $S(10,12,-99,99)$ the inverse is included by

the new algorithm. All bounds for every component are of least significant bit accuracy. The condition number of H^{15} is approximately 10^{22}. Matrices with up to 210 rows were inverted on the UNIVAC 1108. In every case the correctness and non-singularity of the given matrix is verified automatically by the computer.

7. NON-LINEAR SYSTEMS

Consider a function f: $V\mathbb{R}\to V\mathbb{R}$ with continuous first derivative. We desire to find small bounds for regions containing a zero of f. The existence and uniqueness of a zero within the bounds should be verified automatically by the computer. For this purpose consider the following theorem.

Theorem 7.1: Let f: $V\mathbb{R}\to V\mathbb{R}$ be a function with continuous first derivative and let $R\in M\mathbb{R}, \tilde{x}\in V\mathbb{R}$. Denote the Jacobian matrix of f by $f'\in M\mathbb{R}$ and for $X\in \mathbb{I}V\mathbb{R}$ define $f'(X) := \cap\{Y\in \mathbb{I}M\mathbb{R} \mid f'(x)\in Y \text{ for all } x\in X\}$. If then for some $X\in \mathbb{I}V\mathbb{R}$

$$\tilde{x} - R\cdot f(\tilde{x}) + \{I - R\cdot f'(\tilde{x}\cup X)\}\cdot(X - \tilde{x}) \subseteq \overset{\circ}{X}, \tag{7.1}$$

then there exists an $\hat{x}\in\overset{\circ}{X}$ with $f(\hat{x}) = 0$.

Proof: In every ε-neighborhood of a matrix $C\in M\mathbb{R}$ there is a non-singular matrix $\overline{C}\in M\mathbb{R}$ (this can be proved by regarding the determinant of C as a polynomial in n^2 variables which is continuous and not identically vanishing, since all coefficients are ± 1). Therefore according to (7.1) a non-singular matrix $\overline{R}\in M\mathbb{R}$ exists with

$$\tilde{x} - \overline{R}\cdot f(\tilde{x}) + \{I - \overline{R}\cdot f'(\tilde{x}\cup X)\}\cdot(X - \tilde{x}) \subseteq \overset{\circ}{X}. \tag{7.2}$$

Define the function g: $V\mathbb{R}\to V\mathbb{R}$ by

$$x\in V\mathbb{R}: g(x) := x - \overline{R}\cdot f(x). \tag{7.3}$$

g is a function with continuous first derivative. A brief computation using the n-dimensional mean-value theorem yields:

$$\forall x \in X:\ g(x) \in \tilde{x} - \overline{R} \cdot f(\tilde{x}) + \{I - \overline{R} \cdot f'(\tilde{x} \cup X)\} \cdot (X - \tilde{x})$$

(however, for $\tilde{x} \in X$ there is, in general, no $\xi \in \tilde{x} + X$ with $g(x) = \tilde{x} - \overline{R} \cdot f(\tilde{x}) + \{I - \overline{R} \cdot f'(\xi)\} \cdot (X - \tilde{x})$). Therefore by (7.2) and $g(X) := \{g(x) \mid x \in X\}$

$$g(X) \subseteq \overset{\circ}{X}. \tag{7.4}$$

Now by Brouwer's fixed point theorem there is an $\hat{x} \in \overset{\circ}{X}$ with $g(\hat{x}) = \hat{x}$. By the definition (7.3) of g and the non-singularity of $\overline{R}$ this implies $f(\hat{x}) = 0$. □

According to the preceding proof one cannot replace $f'(\tilde{x} \cup X)$ by $\{f'(x) \mid x \in \tilde{x} \cup X\}$ in theorem 7.1. It is not possible to replace $f'(\tilde{x} \cup X)$ by $f'(X)$ as can be demonstrated by counterexamples. Again it is preferable not to include a zero $\hat{x}$ of f itself but the difference between an approximate zero $\tilde{x}$ and $\hat{x}$. Calculating an inclusion of $\hat{x}$ or $\hat{x} - \tilde{x}$ requires the same computing time.

Corollary 7.2: Let $f: V\mathbb{R} \to V\mathbb{R}$ be a function with continuous first derivative and let $R \in M\mathbb{R}$, $\tilde{x} \in V\mathbb{R}$. Define f' and $f'(X)$ for $X \in \mathbb{IV}\mathbb{R}$ as in theorem 7.1. If then for some $X \in \mathbb{IV}\mathbb{R}$ with $0 \in X$

$$-R \cdot f(\tilde{x}) + \{I - R \cdot f'(\tilde{x} + X)\} \cdot X \subseteq \overset{\circ}{X}, \tag{7.5}$$

then there exists an $\hat{x} \in \tilde{x} + \overset{\circ}{X}$ with $f(\hat{x}) = 0$.

The assertions of this corollary can be sharpened under slightly weaker assumptions. To prove the stronger result we need the following lemma.

Lemma 7.3: Let $Z \in \mathbb{IV}\mathbb{R}$, $\mathscr{A} \in LM\mathbb{R}$ and $X \in \mathbb{IV}\mathbb{R}$. If then

$$Z + \mathscr{A} \cdot X \underset{\neq}{\subset} X, \tag{7.6}$$

then for every matrix $A \in \mathscr{A}$ holds $\rho(A) \leq \rho(|A|) < 1$.

Proof: Using (7.6) and (18), p. 153 in [2] we obtain

$$|\mathscr{A}| \cdot d(X) \leq d(\mathscr{A} \cdot X) < d(X).$$

Therefore by corollary 3, p.18 in [42] we have $\rho(|\mathscr{A}|)<1$ and by Perron-Frobenius Theory for every $A\in\mathscr{A}$: $\rho(A)\leq\rho(|A|)\leq\rho(|\mathscr{A}|)<1$. □

A proof for this lemma is given in [34]; the presented proof is due to Alefeld. Next we give a theorem improving Corollary 7.2.

Theorem 7.4: Let f: $V\mathbb{R}\rightarrow V\mathbb{R}$ be a function with continuous first derivative and let $R\in M\mathbb{R}$, $\tilde{x}\in V\mathbb{R}$. Define f' and $f'(X)$ for $X\in\mathbb{I}V\mathbb{R}$ as in theorem 7.1. If then for some $X\in\mathbb{I}V\mathbb{R}$ with $0\in X$

$$-R\cdot f(\tilde{x})+\{I-R\cdot f'(\tilde{x}+X)\}\cdot X\underset{\neq}{\subset}X, \tag{7.7}$$

then the matrix R and each $B\in M\mathbb{R}$ with $B\in f'(\tilde{x}+X)$ is non-singular and there is one and only one $\hat{x}\in\tilde{x}+\overset{\circ}{X}$ with $f(\hat{x})=0$.

Proof: Applying lemma 7.3 to (7.7) yields the non-singularity of R and the matrices $B\in M\mathbb{R}$ with $B\in f'(\tilde{x}+X)$. Define the function g: $V\mathbb{R}\rightarrow V\mathbb{R}$ by

$$x\in V\mathbb{R}\colon g(x):=x-R\cdot f(\tilde{x}+x). \tag{7.8}$$

Then g is continuous and differentiable. As in the proof of theorem 7.1 brief computation using the n-dimensional mean-value theorem yields:

$$\forall x\in X\colon g(x)\in -R\cdot f(\tilde{x})+\{I-R\cdot f'(\tilde{x}+X)\}\cdot X.$$

Then with $g(X):=\{g(x)\mid x\in X\}$ we have $g(X)\underset{\neq}{\subset}X$ and by Brouwer's fixed point theorem there is an $\hat{x}\in\overset{\circ}{X}$ with $g(\hat{x})=\hat{x}$. This implies by (7.8) and the non-singularity of R that $f(\tilde{x}+\hat{x})=0$. Suppose there is a $\hat{y}\in X$ with $f(\tilde{x}+\hat{y})=0$. Then applying the n-dimensional mean-value theorem there is a matrix $B\in f'(\tilde{x}+X)$ with

$$f(\tilde{x}+\hat{y})=f(\tilde{x}+\hat{x})+B\cdot(\tilde{x}+\hat{y}-\tilde{x}-\hat{x}).$$

This implies $B(\hat{y}-\hat{x})=0$ and by the non-singularity of B the theorem is proved. □

The preceding theorem gives under fewer assumptions than in corollary 7.2, in addition the non-singularity of R and every $B \in f'(\tilde{x} + X)$. For the generalization to complex function we need a complex version of the mean-value theorem. This is given in the next lemma, which is due to Böhm (see [6]).

Lemma 7.5: Let $G \in \mathbb{P}V\mathbb{C}$ be convex and non-empty. Then for a holomorphic function $f: G \to \mathbb{C}$ and $z, z_0 \in G$ there are $t_1, t_2 \in \mathbb{R}$ with $0 \leq t_1, t_2 \leq 1$ such that with $\zeta_i := z_0 + t_i(z - z_0)$, $i = 1,2$

$$f(z) = f(z_0) + \mathrm{Re}\,\{f'(\zeta_1) \cdot (z - z_0)\} + j \cdot Im\{f'(\zeta_2) \cdot (z - z_0)\}. \tag{7.9}$$

Remark: f' denotes the gradient of f, Re resp. Im of a vector denotes the vector of real resp. imaginary parts of the components; j is the imaginary unit.

Proof of Lemma 7.5: For $z := x + jy$, $x_0 = z_0 + jy_0$ with $x, y, x_0, y_0 \in V\mathbb{R}$ we have $f(z) = u(x,y) + j \cdot v(x,y)$ with $u, v: V\mathbb{R} \to \mathbb{R}$. By the (real) mean-value theorem there are $t_1, t_2 \in \mathbb{R}$ with $0 \leq t_1, t_2 \leq 1$ such that with $\zeta_i := x_0 + t_i(x - x_0)$ and $\mu_i := y_0 + t_i(y - y_0)$, $i = 1,2$ we have

$$u(x,y) = u(x_0,y_0) + u_x(\xi_1,\mu_1)(x - x_0) + u_y(\xi_1,\mu_1)(y - y_0)$$
$$v(x,y) = v(x_0,y_0) + v_x(\xi_2,\mu_2)(x - x_0) + v_y(\xi_2,\mu_2)(y - y_0).$$

Now short computation using the Cauchy-Riemann differential equations proves the lemma. □

In the following for a function $f: V\mathbb{C} \to V\mathbb{C}$ we denote the Jacobian matrix of f by f'.

Theorem 7.6 (Böhm): Let $z_1, z_2 \in V\mathbb{C}$ and define $Z := \{z \in V\mathbb{C} \mid z_1 \leq z \leq z_2\}$. Let $G \in \mathbb{P}V\mathbb{C}$ with $\tilde{z} \cup Z \subseteq G$ and let $f: G \to V\mathbb{C}$ be a holomorphic function. Define $f'(Z) := \cap\{Y \in \mathbb{M}\mathbb{C} \mid \inf(Y) \leq f(z) \leq \sup(Y) \text{ for } z \in Z\}$. Then for all $z \in Z$

$$f(z) \in f(\tilde{z}) + f'(\tilde{z} \cup Z) \cdot (Z - \tilde{z}). \tag{7.10}$$

Proof: This is a consequence of the preceding lemma. □

According to [36] the assertions of lemma 7.3 remain valid after replacing $\subseteq X$ by $\subseteq \overset{\circ}{X}$ or replacing $\mathbb{R}$ by $\mathbb{C}$. Combining this with the preceding corollary yields

Theorem 7.7: Let $G \in \mathbb{IPV}C$ and let f: $G \to V\mathbb{C}$ be a holomorphic function. Let $R \in M\mathbb{C}$ and $\tilde{z} \in V\mathbb{C}$. Define f' to be the Jacobian matrix of f and define $f'(Z) := \cap\{Y \in \mathbb{IV}\mathbb{C} \mid \inf(Y) \leq f'(z) \leq \sup(Y) \text{ for all } z \in Z\}$ for $Z \in \mathbb{IV}\mathbb{C}$. If then for some $Z \in \mathbb{IV}\mathbb{C}$ with $\tilde{z} + Z \subseteq G$ and $0 \in Z$

$$-R \cdot f(\tilde{z}) + \{I - R \cdot f'(\tilde{z} + Z)\} \cdot Z \subseteq \overset{\circ}{Z}, \tag{7.11}$$

then the matrix R and each matrix $B \in M\mathbb{C}$ with $B \in f'(\tilde{z} + Z)$ is non-singular and there is one and only one $\hat{z} \in \tilde{z} + Z$ with $f(\hat{z}) = 0$.

Proof: Similar to the proof of theorem 7.4.

Both theorem 7.4 and the preceding theorem are applicable on computers as stated in th$ following two corollaries.

Corollary 7.8: Let f: $V\mathbb{R} \to V\mathbb{R}$ be a function with continuous first derivative and let $R \in MS$, $\tilde{x} \in VS$. Let $\diamondsuit$: $V\mathbb{R} \to \mathbb{IV}S$ be a function satisfying $x \in V\mathbb{R} \Rightarrow f(x) \in \diamondsuit(x)$. Define f' to be the Jacobian matrix of f and for $X \in \mathbb{IV}S$ define $f'(X) := \cap\{Y \in \mathbb{IV}S \mid f'(x) \in Y \text{ for all } x \in X\}$. If then for some $X \in \mathbb{IV}S$ with $0 \in X$

$$-R \diamondsuit \diamondsuit(\tilde{x}) \oplus \diamondsuit\{I - R \cdot f'(\tilde{x} \diamondsuit X)\} \diamondsuit X \subsetneq X, \tag{7.12}$$

then the matrix R and each matrix $B \in M\mathbb{R}$ with $B \in f'(\tilde{x} \diamondsuit X)$ is non-singular and there is one and only one $\hat{x} \in \tilde{x} \diamondsuit X$ with $f(\hat{x}) = 0$.

Corollary 7.9: Let $G \in \mathbb{IPV}\mathbb{C}$ and let f: $G \to V\mathbb{C}$ be a holomorphic function. Let $R \in M\mathbb{C}S$ and $\tilde{z} \in V\mathbb{C}S$. Let $\diamondsuit$: $V\mathbb{C} \to \mathbb{IV}\mathbb{C}S$ be a function satisfying $z \in V\mathbb{C} \Rightarrow f(z) \in \diamondsuit(z)$. Define f' to be the Jacobian matrix of f and define $f'(Z) := \cap\{Y \in \mathbb{IV}\mathbb{C}S \mid \inf(Y) \leq f'(z) \leq \sup(Y) \text{ for all } z \in Z\}$ for $Z \in \mathbb{IV}\mathbb{C}S$. If then for some $Z \in \mathbb{IV}S$ with $\tilde{z} + Z \subseteq G$ and $0 \in Z$

$$-R \diamondsuit \diamondsuit(\tilde{z}) \oplus \diamondsuit\{I - R \cdot f'(\tilde{z} \diamondsuit Z)\}) \diamondsuit Z \subseteq \overset{\circ}{Z}, \tag{7.13}$$

then R and each matrix $B \in M\mathbb{C}$ with $B \in f'(\tilde{z} \diamondsuit Z)$ is non-singular and there is one and only one $\hat{z} \in \tilde{z} \diamondsuit Z$ with $f(\hat{z}) = 0$.

Remark: A close reading of the proof of theorem 7.4 yields the non-singularity of each matrix $B \in M\mathbb{IR}$ with $B \in f'(\tilde{x} \diamondsuit X)$ resp. each matrix $B \in M\mathbb{C}$ with $B \in f'(\tilde{z} \diamondsuit Z)$. Moreover, the same proof for the uniqueness of $\hat{x}$ resp. $\hat{z}$ in $\tilde{x} \diamondsuit X$ resp. $\hat{z} \diamondsuit Z$ instead in $\tilde{x} + X$ resp. $\tilde{z} + Z$ can be applied for the preceding two lemmata. □

(7.12) and (7.13) are (effectively) executable on computers according to [21], [24] and [3]. This is true especially for $\diamondsuit \{I - R \cdot f'(\tilde{x} \diamondsuit X)\}$ resp. $\diamondsuit \{I - R \cdot f'(\tilde{z} \diamondsuit Z)\}$. Next we give an algorithm to compute an inclusion of a zero of a system of non-linear equations with automatic verification of existence and uniqueness.

1. Use your favorite floating-point algorithm to compute an approximate zero $\tilde{x}$ of f.

2. Use your favorite floating-point algorithm to compute an approximate inverse of $f'(\tilde{x})$.

3. $Y := ([0])$; $k := 0$; $Z := \diamondsuit(\tilde{x})$; $Z := \ominus R \odot Z$;
   repeat $k := k + 1$; $X := Y \cup 0$; $D := \diamondsuit'(\tilde{x} \diamondsuit X)$;
   $Y := Z \diamondsuit \diamondsuit \{I - R \cdot D\} \diamondsuit X$;
   until $Y \subseteq \overset{\circ}{X}$ or $k > 10$;

4. if $Y \subseteq \overset{\circ}{X}$ then {It has been verified, that there exists one and only one $\hat{x} \in \tilde{x} \diamondsuit \overset{\circ}{X}$ with $f(\hat{x}) = 0$}
   else {No verification}.

Algorithm 7.1. Non-linear Systems of Equations

Here $\diamondsuit: VS \to \mathbb{I}VS$ is any function satisfying $x \in VS \Rightarrow f(x) \in \diamondsuit(x)$ and $\diamondsuit': \mathbb{I}VS \to \mathbb{I}VS$ is a function satisfying $X \in \mathbb{I}VS \Rightarrow \{\forall x \in X$ holds $\diamondsuit'(x) \in \diamondsuit'(X)\}$. There is a similar algorithm for complex systems of non-linear equations and there are similar extensions as in chapter 2, to

functions f: $I\!PV\mathbb{R} \to I\!PV\mathbb{R}$ and f: $I\!PV\mathbb{C} \to I\!PV\mathbb{C}$. In the two latter cases every function f with $f(x) \in f(x)$ for every $x \in \tilde{x} \diamondsuit X$ resp. $\tilde{z} \diamondsuit Z$ has exactly one zero in $\tilde{x} \diamondsuit \overset{\circ}{X}$ resp. $\tilde{z} \diamondsuit \overset{\circ}{Z}$ (in the terminology of corollaries 7.8 and 7.9).

The above algorithm can be used after a floating-point algorithm to determine the accuracy of the computed approximation. The computing time is $(k+1)n^3$ plus one evaluation of $\diamondsuit$ and k evaluations of $\diamondsuit'$, where the computing time for evaluating $\diamondsuit$ resp. $\diamondsuit'$ is roughly the same as for f resp. f'. As shown in the following examples the algorithm terminates almost always with $k = 1$. This automatic verification process could replace efforts on the part of the user to make an approximation plausible. The question, for which initial X algorithms 7.1 terminates is answered by lemma 2.9.

Theorem 7.4 resp. theorem 7.7 can be regarded as an extension of the Kantorovish Lemma.

Experience showed, that if the approximation $\tilde{x}$ is of the magnitude of a solution of the non-linear system, then algorithm 7.1 terminates for $k \leq 1$ with results of least significant bit accuracy.

The following computational results are from the UNIVAC 1108 at the University of Karlsruhe. Here the floating-point screen S is (2,27,-128,127). So the mantissa length is approximately $8\frac{1}{2}$ decimal digits. We treated Example 7 in [1], Problem 1 in [31] and Problem 2 in [31]. For more examples see [36]. In the following table we display from left to right

- the number of the problem
- n: number of functions and variables
- Newton-steps: number of Newton iterations starting with the initial guess prescribed in the cited literature
- k: defined in step 3 of algorithm 7.1
- succeeded: yes indicates $Y \subseteq \overset{\circ}{X}$ in step 3 of algorithm 7.1

- digits guaranteed: least number of digits for which the left and right bound coincide; here an additional l.s.b.a. means least significant bit accuracy, i.e. that the left and right bound of the inclusion of each component are consecutive numbers in the floating-point screen.

Example 7 [1]:

Discretization of $3\ddot{y}y + \dot{y}^2 = 0$, $y(0) = 0$, $y(1) = 20$.

$$f_1 = 3x_1(x_2 - 2x_1) + x_2^2/4$$
$$f_i = 3x_i(x_{i+1} - 2x_i + x_{i-1}) + (x_{i+1} - x_{i-1})^2/4 \qquad 2 \le i \le n-1$$
$$f_n = 3x_n(20 - 2x_n + x_{n-1}) + (20 - x_{n-1})^2/4$$

Solution $10t^{3/4}$; initial guess $x_i = 10$ for $i \le i \le n$.

Example 1 in [31]:

Discretization of $u''(t) = \frac{1}{2}(\bar{u}(t) + t + 1)^3$, $0 < t < 1$, $\bar{u}(0) = \bar{u}(1) = 0$

$$x_k = \bar{u}(t_k),\ f_k(x) \equiv 2x_k - x_{k-1} + \frac{1}{2}h^2(x_k + t_k + 1)^3 \qquad 1 \le k \le n$$
$$x_0 = x_{k+1} = 0,\ t_k = k \cdot h;\ h = (n+1)^{-1}$$

initial guess $x \equiv (\xi_i)$, $\xi_i = t_i(t_i - 1)$ $\qquad 1 \le i \le n$.

Example 2 in [31]:

$$\bar{u}(t) + \int_0^1 H(s,t)(\bar{u}(s) + s + 1)^3 ds = 0$$

$$H(s,t) = \begin{cases} s(1-t) & s \le t \\ t(1-s) & s > t \end{cases}$$

$$x_k = \bar{u}(t_k),\ f_n(x) \equiv x_n + \frac{1}{2}\left\{(1 - t_k)\sum_{j=i}^{k} t_j(x_j + t_j + 1)^3 + t_k \sum_{j=k+1}^{n} 1 - t_j)(x_j + t_j + 1)^2\right\}$$

$x_0 = x_{n+1} = 0$, $t_j = jh$; $h = (n+1)^{-1}$; initial guess $x_i = t_i(t_i - 1)$.

Problem	n	Newton – steps	k	succeeded	digits guaranteed
1)	10	8	2	yes	$8\frac{1}{2}$ (l.s.b.a.)
	20	6	2	yes	$8\frac{1}{2}$ (l.s.b.a.)
	50	6	3	yes	8
	100	7	3	yes	8
2)	10	4	1	yes	$8\frac{1}{2}$ (l.s.b.a.)
	20	6	1	yes	$8\frac{1}{2}$ (l.s.b.a.)
	50	6	1	yes	8
3)	10	3	1	yes	$8\frac{1}{2}$ (l.s.b.a.)
	20	4	1	yes	$8\frac{1}{2}$ (l.s.b.a.)
	50	4	1	yes	8

The additional computing time required to obtain results with verification of correctness is about k times the computing time for one usual floating-point Newton iteration. However, any recomputing with slightly altered entries to gain in security is unnecessary.

For further improvements of the above algorithm with a reduction of k to 1 in every of the mentioned examples see chapter 11 of this article.

8. THE ALGEBRAIC EIGENVALUE PROBLEM

The eigenproblem (cf. [34]) can be formulated as a non-linear system. For $A \in M\mathbb{R}$ let

$$\begin{aligned} Ax - \lambda x &= 0 \\ e'_k x - \zeta &= 0. \end{aligned} \tag{8.1}$$

Here e'_k is the k^{th} unitvector, $1 \le k \le n$. If $\zeta \neq 0$ then any pair (x,λ) with $x \in V\mathbb{R}$, $\lambda \in \mathbb{R}$ is an eigenvector/eigenvalue pair of A. In the following the proofs first given in [34] are shortened and completed by lemma 8.3. Finally theorem 8.8 is added. From the preceding chapter a theorem can be derived to compute an inclusion for an eigenvector/eigenvalue pair of A satisfying (8.1):

Theorem 8.1: Let $A\in M_n\mathbb{R}$, $R\in M_{n+1}\mathbb{R}$, $\tilde{x}\in V_n\mathbb{R}$ and $\tilde{\lambda},\zeta\in\mathbb{R}$ with $\zeta\neq 0$. Define for $Y\in\mathbb{IV}_n\mathbb{R}$, $M\in\mathbb{IR}$ the function G: $\mathbb{IV}_{n+1}\mathbb{R}\to\mathbb{PV}_{n+1}\mathbb{R}$ by

$$G\begin{pmatrix} Y \\ M \end{pmatrix} := \begin{pmatrix} \tilde{x} \\ \tilde{\lambda} \end{pmatrix} - R\cdot\begin{pmatrix} A\tilde{x}-\tilde{\lambda}\tilde{x} \\ e'_k\tilde{x}-\zeta \end{pmatrix} + \left\{ I_{n+1} - R\cdot\begin{pmatrix} A-(\tilde{\lambda}\underline{\cup}M)I_n & -(\tilde{x}\underline{\cup}Y) \\ e'_k & 0 \end{pmatrix}\right\}\cdot\begin{pmatrix} Y-\tilde{x} \\ M-\tilde{\lambda} \end{pmatrix}. \tag{8.2}$$

If then for some $X\in\mathbb{IV}_n\mathbb{R}$, $\Lambda\in\mathbb{IR}$

$$G(T)\subseteq\overset{\circ}{T} \text{ for } T=\begin{pmatrix} X \\ \Lambda \end{pmatrix} \tag{8.3}$$

holds, then there exists one and only one eigenvector/eigenvalue pair $(\hat{x},\hat{\lambda})$ with $\hat{x}\in X$ and $\hat{\lambda}\in\Lambda$.

Here I_n is the $n\times n$ unit matrix and $\tilde{\lambda}\underline{\cup}M := \square(\tilde{\lambda}\cup M)$, $\tilde{x}\underline{\cup}Y := \square(\tilde{x}\cup Y)$ as defined in chapter 1. There are similar extensions to complex eigenvector/eigenvalue problems and problems with uncertain data. Next we will improve the assertions of theorem 8.1 under weaker assumptions. Instead of the Jacobian matrix used in (8.2), consider

$$S(X) := \begin{pmatrix} A-\tilde{\lambda}I_n & -X \\ e'_k & 0 \end{pmatrix} \in\mathbb{PM}_{n+1}\mathbb{R} \text{ with } X\in\mathbb{IV}_n\mathbb{R}. \tag{8.4}$$

Define

$$Z := \begin{pmatrix} \tilde{x} \\ \tilde{\lambda} \end{pmatrix} - R\cdot\begin{pmatrix} A\tilde{x}-\tilde{\lambda}\tilde{x} \\ e'_k\tilde{x}-\zeta \end{pmatrix} \in\mathbb{PV}_{n+1}\mathbb{R}. \tag{8.5}$$

We will show that using the function G^*: $\mathbb{IV}_{n+1}\mathbb{R}\to\mathbb{PV}_{n+1}\mathbb{R}$ defined by

$$G^*\begin{pmatrix} Y \\ M \end{pmatrix} := Z + (I_{n+1} - R \cdot S(Y)) \cdot \begin{pmatrix} Y - \tilde{x} \\ M - \tilde{\lambda} \end{pmatrix} \tag{8.6}$$

instead of the G in theorem 8.1, it follows from (8.3) that there exists exactly one eigenvector $\hat{x}$ of A in X, there exists exactly one eigenvalue $\hat{\lambda}$ of A in Λ, $A\hat{x} = \hat{\lambda}\hat{x}$ holds and that $\hat{\lambda}$ is of multiplicity one. Obviously

$$G^*\begin{pmatrix} Y \\ M \end{pmatrix} \subseteq G\begin{pmatrix} Y \\ M \end{pmatrix},$$

and thus the assumptions are weaker.

Lemma 8.2: Define G^*: $\mathbb{IV}_{n+1}\mathbb{R} \to \mathbb{IP}V_{n+1}\mathbb{R}$ by (8.6) for $R \in M_{n+1}\mathbb{R}$; $\tilde{x} \in V_n\mathbb{R}$; $\tilde{\lambda}, \zeta \in \mathbb{R}$; $\zeta \neq 0$.

If

$$G^*(T) \subseteq \overset{\circ}{T} \text{ with } T = \begin{pmatrix} X \\ \Lambda \end{pmatrix} \text{ for some } X \in \mathbb{IV}_n\mathbb{R} \text{ and } \Lambda \in \mathbb{IR}, \tag{8.7}$$

then there exists an eigenvector $\hat{x}$ of A with $\hat{x} \in X$ and an eigenvalue $\hat{\lambda}$ of A with $\hat{\lambda} \in \Lambda$ and

$$A\hat{x} = \hat{\lambda}\hat{x}. \tag{8.8}$$

Proof: Define f: $V_{n+1}\mathbb{R} \to V_{n+1}\mathbb{R}$ by

$$\begin{aligned} f\begin{pmatrix} x \\ \lambda \end{pmatrix} &:= \begin{pmatrix} x \\ \lambda \end{pmatrix} - R \cdot \begin{pmatrix} Ax - \lambda x \\ e'_k x - \zeta \end{pmatrix} \\ &= Z + \left\{ I_{n+1} - R \cdot \begin{pmatrix} A - \tilde{\lambda} I_n & -x \\ e'_k & 0 \end{pmatrix} \right\} \cdot \begin{pmatrix} x - \tilde{x} \\ \lambda - \tilde{\lambda} \end{pmatrix}. \end{aligned} \tag{8.9}$$

Then

$$y\in Y,\ \mu\in M\Rightarrow f\begin{pmatrix} y \\ \mu \end{pmatrix}\in G^*\begin{pmatrix} Y \\ M \end{pmatrix}. \tag{8.10}$$

By (8.9), (8.7) and the fixed point theorem of Brouwer there is a $(\hat{x},\hat{\lambda})\in(X,\Lambda)$ with $f(\hat{x},\hat{\lambda}) = (\hat{x},\hat{\lambda})$, and by (8.9)

$$\begin{pmatrix} A\hat{x}-\hat{\lambda}\hat{x} \\ e'_k x-\zeta \end{pmatrix}\in \ker\ R.$$

By lemma 7.3, R is non-singular and using $e'_k\hat{x} = \zeta\neq 0$ the proof is completed. □

Our aim is to prove the uniqueness of $\hat{x}$ in X and $\hat{\lambda}$ in Λ separately. To this end we first derive the uniqueness of the pair $(\hat{x},\hat{\lambda})$ in (X,Λ).

Lemma 8.3: With the assumptions of lemma 8.2 the eigenvector/eigenvalue pair $(\hat{x},\hat{\lambda})$ is uniquely determined in (X,Λ).

Proof: Define f_z: $V_{n+1}\mathbb{R}\to V_{n+1}\mathbb{R}$ by

$$f_z\begin{pmatrix} w \\ \sigma \end{pmatrix} := \begin{pmatrix} w \\ \sigma \end{pmatrix} - R\cdot\begin{pmatrix} Aw-\tilde{\lambda}w-\sigma z+\tilde{\lambda}z \\ e'_k w-\zeta \end{pmatrix}. \tag{8.11}$$

Then short computation yields for arbitrary $x\in V_n\mathbb{R}$

$$f_z\begin{pmatrix} w \\ \sigma \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ \tilde{\lambda} \end{pmatrix} - R\cdot\begin{pmatrix} A\tilde{x}-\tilde{\lambda}\tilde{x} \\ e'_k\tilde{x}-\zeta \end{pmatrix} + \left\{I_{n+1} - R\cdot\begin{pmatrix} A-\tilde{\lambda}I_n & -z \\ e'_k & 0 \end{pmatrix}\right\}\cdot\begin{pmatrix} w-\tilde{x} \\ \sigma-\tilde{\lambda} \end{pmatrix}.$$

From (8.7) and (8.12) follows for every $z\in X$

$$f_z(T)\subseteq \overset{\circ}{T} \text{ with } T = \begin{pmatrix} X \\ \Lambda \end{pmatrix}. \tag{8.13}$$

Therefore for every $z \in X$ there is a fixed point of f_z in $\overset{\circ}{T}$. Obviously (x,λ) is a fixed point of f_x.

Lemma 7.3 imples the non-singularity of every matrix

$$M_z := \begin{pmatrix} A - \tilde{\lambda} I_n & -z \\ e'_k & 0 \end{pmatrix} \in M_{n+1}\mathbb{R} \quad \text{for every } z \in X. \tag{8.14}$$

For the purpose of establishing a contradiction we assume $Ax = \lambda x$ and $Ay = \mu y$ with $x,y \in X$ and $\lambda,\mu \in \Lambda$ with $\lambda \neq \mu$. Suppose further $\tilde{\lambda}$ is an eigenvalue of A with eigenvector v. Then in case $e'_k \cdot v = 0$ we have $(v,0)' \in \ker M_x$, in case $e'_k \cdot v \neq 0$ w.l.o.g. $e'_k \cdot v = \zeta$ and $(v - x, \tilde{\lambda} - \lambda)' \in \ker M_x$ contradicting (8.14). Therefore especially $\tilde{\lambda} \neq \lambda$ and $\tilde{\lambda} \neq \mu$. Next we will compute a fixed point of $f_{x+\delta(y-x)}$. Short computation yields for $\delta \neq (\mu - \tilde{\lambda})/(\mu - \lambda)$

$$\begin{aligned} f_{x+\delta(y-x)} \begin{pmatrix} w \\ \sigma \end{pmatrix} = \begin{pmatrix} w \\ \sigma \end{pmatrix} \text{ for } & w(\delta) = x + \frac{\delta(\lambda - \tilde{\lambda})}{N}(y - x) \text{ and} \\ & \sigma(\delta) = \tilde{\lambda} + (\lambda - \tilde{\lambda})(\mu - \tilde{\lambda})/N \text{ with} \\ & N = (1-\varepsilon)(\mu - \tilde{\lambda}) + \varepsilon(\lambda - \tilde{\lambda}). \end{aligned} \tag{8.15}$$

This fixed point $(w(\delta), \sigma(\delta))'$ lies therefore on the straight line connecting x and y. By (8.7) and $x,y \in X \setminus \partial X$ follows the existence of $\delta_1, \delta_2 \in \mathbb{R}$ with

$$\begin{aligned} \delta_1 < 0 < 1 < \sigma_2 \text{ and } \delta_1 \leq \delta \leq \delta_2 \Rightarrow & x + \delta(y - x) \in X \\ \text{and } & x + \delta_1(y - x) \in \partial X,\ x - \delta_2(y - x) \in \partial X. \end{aligned} \tag{8.16}$$

For every $\delta \in [\delta_1, \delta_2]$ with $\delta \neq (\mu - \tilde{\lambda})/(\mu - \lambda)$ we have $N \neq 0$ and by (8.13)

$$w(\delta) \in \overset{\circ}{X}. \tag{8.17}$$

If $(\mu - \tilde{\lambda})/(\mu - \lambda) \in [\delta_1, \delta_2]$ then there exists $w(\delta) \exists \overset{\circ}{X}$ contradicting (8.17). Therefore (8.17) and (8.18) implies

$$\delta \in [\delta_1, \delta_2] \Rightarrow \delta_1 < \frac{\delta(\lambda - \tilde{\lambda})}{(1-\delta)(\mu - \tilde{\lambda}) + \delta(\lambda - \tilde{\lambda})} < \delta_2. \tag{8.18}$$

Suppose $N>0$. Then the left inequality with $\delta = \delta_1$ implies $\mu>\lambda$, the right inequality with $\delta = \delta_2$ implies $\tilde{\lambda}>\mu$. There is the same contradiction for $N<0$. For $\lambda = \mu$ we have $N = \lambda-\tilde{\lambda}\neq 0$, $w(\delta) = x + \delta(y - x)$ and $\sigma(\delta) = \lambda$ demonstrating lemma 8.3. □

Lemma 8.4: Under the assumptions of lemma 8.3 given an eigenvalue μ of A with $\mu\in\Lambda$ every eigenvector y of A corresponding to μ must in X.

Proof: Suppose $Ay = \mu y$ with $\mu\in\Lambda$. Given f from (8.9) we define g_μ: $V_n\mathbb{R}\to V_n\mathbb{R}$ to be the first n components of $f(t,\mu)$:

$$t\in V_n\mathbb{R} \Rightarrow f\begin{pmatrix} t \\ \mu \end{pmatrix} = \begin{pmatrix} g_\mu(t) \\ \nu \end{pmatrix} \text{ with some } \nu\in\mathbb{R}.$$

According to (8.10 and (8.7), g_μ is a continuous self-mapping of X:

$$g_\mu: X\to X \text{ and } g_\mu(X)\subseteq X\setminus\partial X. \tag{8.19}$$

Brouwer's fixed point theorem states that

$$\exists z\in X: g_\mu(z) = z, \text{ i.e. } f\begin{pmatrix} z \\ \mu \end{pmatrix} = \begin{pmatrix} z \\ \mu^* \end{pmatrix} \tag{8.20}$$

for some $\mu^*\in\mathbb{R}$. Define g: $V_{n+1}\mathbb{R}\to V_{n+1}\mathbb{R}$ by

$$w\in V_n\mathbb{R},\ \sigma\in\mathbb{R} \Rightarrow g\begin{pmatrix} w \\ \sigma \end{pmatrix} := Z + \left\{ I_{n+1} - R\cdot\begin{pmatrix} A-\sigma I_n & -\tilde{x} \\ e'_k & 0 \end{pmatrix}\right\}\cdot\begin{pmatrix} w-\tilde{x} \\ \sigma-\tilde{\lambda} \end{pmatrix},$$

then $g(w,\sigma) = f(w,\sigma)$ and (8.10) holds with g substituted for f. Applying lemma 7.3 yields that

$$T(\sigma) := \begin{pmatrix} A-\sigma I_n & -\tilde{x} \\ e'_k & 0 \end{pmatrix} \text{ is non-singular for every } \sigma\in\Lambda.$$

Since y is an eigenvector of A, $y\neq 0$. So $T(\mu)\cdot(y,0)\neq 0$ yields $y_k := e'_k\cdot y\neq 0$. Define

$$M(\nu) := \left\{ t \in V_n\mathbb{R} \mid f\begin{pmatrix} t \\ \mu \end{pmatrix} = \begin{pmatrix} t \\ \nu \end{pmatrix} \right\} \quad \text{for} \quad \nu \in \mathbb{R}. \tag{8.21}$$

For the purpose of establishing a contradiction we assume $\mu \neq \mu^*$ and define h: $\mathbb{R} \to V_n\mathbb{R}$ by

$$\nu \in \mathbb{R} \Rightarrow h(\nu) := \zeta z + \eta y \quad \text{with} \quad \xi := (\mu - \nu)(\mu - \mu^*)^{-1},\ \eta = \zeta(1-\xi)y_k^{-1}. \tag{8.22}$$

Then brief computation using (8.20) and (8.9) yields

$$h(\nu) \in M(\nu) \quad \text{for every} \quad \nu \in \mathbb{R}. \tag{8.23}$$

h is continuous and

$$h(\nu + \varepsilon) - h(\nu) = \varepsilon \bullet \{\zeta(\mu - \mu^*)^{-1} y_k^{-1} \bullet y - (\mu - \mu^*)^{-1} \bullet z\}. \tag{8.24}$$

If $z = \zeta y_k^{-1} \bullet y$ then $(A - \mu I)z = 0$ and $e'_k z = \zeta$ which contradicts $\mu \neq \mu^*$ according to (8.20) and (8.9). Thus $z \neq \zeta y_k^{-1} \bullet y$ and from (8.24) follows

$$|h(\nu)| \to \infty \quad \text{for} \quad \nu \to \infty. \tag{8.25}$$

For every $\nu \in \mathbb{R}$, any $t \in M(\nu)$ is a fixed point of g_μ. (8.20) yields $z = h(\mu^*) \in M(\mu^*)$ and $z \in X$, so by (8.25) there exists a fixed point of g_μ on ∂X. This contradicts (8.19).

Thus, $\mu = \mu^*$ and the lemma is proved. □

Lemma 8.5: Under the assumptions of lemma 8.2 given an eigenvector y of A with $y \in X$ the eigenvalue μ of A corresponding to y must lie in Λ.

Proof: Suppose $Ay = \mu y$ with $y \in X$. Using f from (8.9) we define g_y: $\mathbb{R} \to \mathbb{R}$ be the $(n+1)^{\text{-st}}$ component of $f(y,\nu)$:

$$\nu \in \mathbb{R} \Rightarrow f\begin{pmatrix} y \\ \nu \end{pmatrix} = \begin{pmatrix} z \\ g_y(\nu) \end{pmatrix} \quad \text{with some} \quad z \in V_n\mathbb{R}. \tag{8.26}$$

According to (8.10) and (8.7) g_y is a continuous self-mapping of Λ:

$$g_y\colon \Lambda\to\Lambda \quad \text{and} \quad g_y(\Lambda)\subseteq\Lambda\setminus\partial\Lambda. \tag{8.27}$$

Brouwer's fixed point theorem states

$$\exists\sigma\in\Gamma\colon g_y(\sigma)=\sigma, \text{ i.e. } f\begin{pmatrix} y \\ \sigma \end{pmatrix} = \begin{pmatrix} z \\ \sigma \end{pmatrix} \tag{8.28}$$

for some $z\in V_f\mathbb{R}$. Define

$$M(t):=\left\{\nu\in\mathbb{R}\mid f\begin{pmatrix} y \\ \nu \end{pmatrix} = \begin{pmatrix} t \\ \nu \end{pmatrix}\right\} \quad \text{for} \quad t\in V_n\mathbb{R}. \tag{8.29}$$

Assuming $\mu\neq\sigma$ we define

$$\mathrm{v}:= z + \eta(y - z) \quad \text{with} \quad \eta:= (\nu-\sigma)/(\mu-\sigma).$$

A brief computation using (8.28) and (8.29) yields $\sigma\in M(z)$ and $\nu\in M(\mathrm{v})$ for every $\nu\in\mathbb{R}$, whence for every $\nu\in\mathbb{R}$ there is a $\mathrm{v}\in V_n\mathbb{R}$ with $\nu\in M(\mathrm{v})$. By (8.29) and (8.26) every ν in some $M(\mathrm{v})$ is a fixed point of g_y. This contradicts (8.27). Thus $\mu = \sigma$ and by (8.28), $\mu\in\Lambda$. □

After these preparatory lemmata, we are ready to state the following theorem.

Theorem 8.6: Define G^*: $\mathbb{I}V_{n+1}\mathbb{R}\to\mathbb{I}\!P V_{n+1}\mathbb{R}$ by (8.6) for $R\in M_{n+1}\mathbb{R}$; $\tilde{x}\in V_n\mathbb{R}$; $\tilde{\lambda},\zeta\in\mathbb{R}$; $\zeta\neq 0$.

If

$$G^*(T)\subseteq \overset{\circ}{T} \text{ with } T=\begin{pmatrix} X \\ \Lambda \end{pmatrix} \text{ for some } X\in\mathbb{I}V_n\mathbb{R},\ \Lambda\in\mathbb{I}\mathbb{R}$$

then all of the following are true:

1) there is one and only one eigenvector $\hat{x}$ of A with $\hat{x}\in X$,
2) there is one and only one eigenvalue $\hat{\lambda}$ of A with $\hat{\lambda}\in\Lambda$,
3) these are corresponding, i.e. $A\hat{x} = \hat{\lambda}\hat{x}$ and
4) the multiplicity of $\hat{\lambda}$ is one.

Proof: The existence of an eigenvector/eigenvalue pair $(\hat{x},\hat{\lambda})$ in (X,Λ) follows by lemma 8.2. Hence we have 3) and 1) and 2) follow by lemmata 8.4 and 8.3 resp. lemmata 8.5 and 8.3. Assertion 4) follows by lemma 8.3. □

In contrast to theorem 8.1, we use $\tilde{\lambda}$ instead of $\tilde{\lambda}\underline{\cup}\Lambda$ and $-X$ instead of $-(\tilde{x}\underline{\cup}X)$, which reduces the diameter of $G^*(T)$ compared with $G(T)$ significantly. Theorem 8.6 is applicable on computers. This is shown by the following theorem. We formulate it at once for finding an inclusion of the difference between the solution $(\hat{x},\hat{\lambda})$ and an approximate solution $(\tilde{x},\tilde{\lambda})$.

Theorem 8.7: Let $A\in M_n S$, $R\in M_{n+1}S$, $\tilde{x}\in V_n S$ and $\tilde{\lambda},\zeta\in S$ with $\zeta\neq 0$. For $X\in \mathbb{IV}_n S$ define

$$Q(X) := \begin{pmatrix} A-\tilde{\lambda}I_k & -\tilde{x}-X \\ e'_k & 0 \end{pmatrix} \in \mathbb{IM}_{n+1}\mathbb{R} \quad \text{and}$$
$$Z := \diamondsuit R \odot \diamondsuit \begin{pmatrix} A\tilde{x}-\tilde{\lambda}\tilde{x} \\ e'_k\tilde{x}-\zeta \end{pmatrix} \in \mathbb{IV}_{n+1}S. \tag{8.30}$$

If then for some $X\in \mathbb{IV}_n S$ and $\Lambda\in \mathbb{IR}$

$$Z \diamondsuit \diamondsuit \{I_{n+1} - R\bullet Q(X)\} \diamondsuit T \subseteq \overset{\circ}{T} \text{ with } T := \begin{pmatrix} X \\ \Lambda \end{pmatrix}, \tag{8.31}$$

then the matrix R and each matrix $B\in M_{n+1}\mathbb{R}$ and $B\in Q(X)$ is non-singular and the following are true:

1) there is one and only one eigenvector $\hat{x}$ of A with $\hat{x}\in\tilde{x} + X$,
2) there is one and only one eigenvalue $\hat{\lambda}$ of A with $\hat{\lambda}\in\tilde{\lambda} + \Lambda$,
3) they are corresponding, i.e. $A\hat{x} = \hat{\lambda}\hat{x}$ and
4) the multiplicity of $\hat{\lambda}$ is one.

Proof: This is a consequence of theorem 8.6. □

The uniqueness of $\hat{x}$ resp. $\hat{\lambda}$ cannot be guaranteed in $\tilde{x} \diamondsuit X$ resp. $\tilde{\lambda} \diamondsuit \Lambda$. (8.31) and especially $\diamondsuit(\square\{I - R\bullet Q(X)\})$ is (effectively) executable on computers using the precise scalar product (cf. [3]).

Theorem 8.7 yields an algorithm to include an eigenvector/eigenvalue pair of a given matrix A with automatic verification of correctness. For the algorithm and further improvements see [34]. Of course in the actual implementation x_k need not to be stored as an additional variable. Instead (8.30) and (8.31) are rewritten in n variables. It is possible to insert some $X^* \in \mathbb{IV}_n S$, $\Lambda^* \in \mathbb{IS}$ with $X^* \supseteq X$, $\Lambda^* \supseteq \Lambda$ in (8.30), (8.31). If for both for (X,Λ) and (X^*,Λ^*) the condition (8.31) is satisfied, then it has been verified that there is no eigenvalue of A in $\Lambda^* \setminus \Lambda$. The computing time for the algorithm is approximately $2n^3$. Each additional evaluation of (8.31) with another T costs $\sim 3n^2$.

Finally we mention another version of theorem 8.7.

Theorem 8.8: Let $A \in M_n S$, $R \in M_{n+1} S$, $\tilde{x} \in V_n S$ and $\tilde{\lambda}$, $\zeta \in S$ with $\zeta \neq 0$. If the linear system $Cx = b$, $b \in \mathbb{IB}_n S$ with

$$C := \begin{pmatrix} A - \tilde{\lambda} I_n & -\tilde{x} \\ e'_k & 0 \end{pmatrix} \quad \text{and} \quad b := \Diamond \begin{pmatrix} -A\tilde{x} + \tilde{\lambda}\tilde{x} + \Lambda X \\ 0 \end{pmatrix}$$

is solved using algorithm 2.1 yielding an inclusion of the solution $(Y,M)'$, $Y \in \mathbb{IV}_n S$, $M \in \mathbb{IS}$ and, moreover,

$$Y \subseteq X \quad \text{and} \quad M \subseteq \Lambda$$

is satisfied, then all assertions of theorem 8.7 remain valid.

Remark: Algorithm 2.1 has to be used in its version for interval right hand side b as described in chapter 2. In general, $A - \tilde{\lambda} I_n$ is not an element of $M_{n+1} S$. However, algorithm 2.1 can be applied for point matrices, for instance, by splitting a product $(A - \tilde{\lambda} I_n)x$, $x \in V_n S$ in a scalar product of length $n + 1$. This assures that all assertions respecting algorithm 2.1 remain true.

Proof of theorem 8.8: A brief computation using (2.2) yields exactly that provision (8.30) in theorem 8.7 is satisfied. Therefore all assumptions of theorem 8.7 are valid. □

Theorem 8.8 extends Satz 3.7 in [47]. In this specific case we do not assume $0 \in X$, $0 \in \Lambda$

but conclude (extending the cited Satz 3.7) the non-singularity of C,the uniqueness of the eigenvector/eigenvalue pair, the uniqueness of the eigenvalue and that the multiplicity of the eigenvalue is 1. Most important is the fact that the non-singularity of C is verified by the computer and not assumed to be checked by the user (which is, in fact, hardly solvable). This makes the corresponding algorithm widely applicable especially for non-mathematicians.

There are similar extensions, as in chapter 7, to complex matrices and problems with uncertain data. In the latter case (with a matrix $\mathscr{A} \in \mathbb{I}M_n S$) the assertions 1), 2), 3) and 4) remain valid for any $A \in M_n \mathbb{R}$ with $A \in \mathscr{A}$.

The following numerical examples were computed on the UNIVAC 1108 at the University of Karlsruhe. We denote by

H	the Hilbert-matrix
P	the Pascal-matrix
S_1, S_2	matrices with uniformly distributed eigenvalues in $[-1,1]$, $[1,10]$, resp.
C	a matrix with clustered eigenvalues $1 + i \cdot 10^{-5}$, $i = 1(1)n$
R	a randomly generated matrix with $\lvert R_{ij} \rvert \leq 1$.

We first applied a built-in procedure to compute approximations $\tilde{x}, \tilde{\lambda}$ for the eigenvectors and eigenvalues of A, resp. Then the new algorithm was applied. The following table displays the matrix, the number of rows n, the maximum relative error $\tilde{\Delta}$ of the components of all approximations $\tilde{x}, \tilde{\lambda}$ and the number of digits guaranteed in the final inclusions for all eigenvectors and all eigenvalues of the matrix. Here an additional l.s.b.a. indicates that all components of the inclusion of eigenvectors and the inclusions of eigenvalues are of least significant bit accuracy, i.e. left and right bounds are consecutive numbers in the floating-point screen $S = (2,27, -128,127)$. The * for $\tilde{\Delta}$ in H^7 indicates, that the approximation $\tilde{\lambda} = -2.17_{10^{-3}}$ for one eigenvalue was of wrong sign (the correct value is $+1.259..._{10^{-3}}$).

matrix	n	$\tilde{\Delta}$	digits guaranteed
H	6	$1.7_{10^{-2}}$	$8\frac{1}{2}$ (l.s.b.a.)
	7	*	$8\frac{1}{2}$ (l.s.b.a.)
	8	43	$8\frac{1}{2}$ (l.s.b.a.)
P	8	$1.5_{10^{-3}}$	$8\frac{1}{2}$ (l.s.b.a.)
	9	$3.0_{10^{-3}}$	$8\frac{1}{2}$ (l.s.b.a.)
S_1	20	$5.6_{10^{-2}}$	$8\frac{1}{2}$ (l.s.b.a.)
S_2	20	$3.1_{10^{-2}}$	$8\frac{1}{2}$ (l.s.b.a.)
C	20	$1.9_{10^{-1}}$	8
R	50	$6.9_{10^{-6}}$	$8\frac{1}{2}$ (l.s.b.a.)

9. REAL AND COMPLEX ZEROS OF POLYNOMIALS

Consider a polynomial p of degree n. p can be regarded as a (continuously differentiable) mapping, so the theorems and corollaries derived in chapter 7 are applicable. Here we mention two theorems for real zeros of real polynomials and complex zeros of complex polynomials. They both are formulated directly for application on the computer, for an inclusion between the difference of a zero and an approximation.

Theorem 9.1: Let $p(x) = \sum_{i=0}^{n} a_i \cdot x^i$ with $a_i \in S$ for $0 \leq i \leq n$ and let $\tilde{x} \in S$, $r \in S$ be given.
Let $\diamondsuit_p : S \to \mathbb{I}S$ resp. $\diamondsuit_p' : \mathbb{I}S \to \mathbb{I}S$ be functions satisfying $x \in S \Rightarrow p(x) \in \diamondsuit_p(x)$ resp. $p'(x) \in \diamondsuit_p'(x)$. If then for some $X \in \mathbb{I}S$ with $0 \in X$

$$\ominus r \diamond \diamondsuit_p(\tilde{x}) \oplus \square \{1 - r \cdot \diamondsuit_p'(\tilde{x} \oplus X)\} \diamond X \subsetneq X, \tag{9.2}$$

then there exists one and only one $\hat{x} \in \mathbb{R}$ with $\hat{x} \in \tilde{x} \oplus X$ and $p(\hat{x}) = 0$. $\hat{x}$ is a simple zero of p.

Theorem 9.2: Let $p(z) = \sum_{i=0}^{n} c_i z^i$ with $c_i \in \mathbb{C}S$ for $0 \leq i \leq n$ and let $\tilde{z} \in \mathbb{C}S$, $r \in \mathbb{C}S$.
Let $\diamondsuit_p : \mathbb{C}S \to \mathbb{I}\mathbb{C}S$ resp. $\diamondsuit_p' : \mathbb{I}\mathbb{C}S \to \mathbb{I}\mathbb{C}S$ be functions satisfying $x \in \mathbb{C}S \Rightarrow p(x) \in \diamondsuit_p(x)$ resp. $\diamondsuit_p'(x) \in \diamondsuit_p'(x)$. If then for some $Z \in \mathbb{I}\mathbb{C}S$ with $0 \in Z$

$$\ominus r \diamond \diamondsuit_p(\tilde{z}) \text{ i} \oplus \square \{1 - r \cdot \diamondsuit_p'(\tilde{z} \oplus Z)\} \diamond Z \subsetneq Z, \tag{9.2}$$

then there exists one and only one $\hat{z}\in\mathbb{C}$ with $\hat{z}\in\tilde{z}$ $\diamondsuit$ Z and $p(\hat{z}) = 0$. $\hat{z}$ is a simple zero of p.

The *proofs* are an immediate consequence of corollaries 7.8 and 7.9. However, both theorems can be proved directly using Banach's fixed point Theorem.

The functions $\diamondsuit_{(p)}$ resp. $\diamondsuit_{(p)}'$ may be the usual interval extensions of p resp. p'. However, this is an overestimation. In chapter 11 a new method will be derived for the computation of the value of arbitrary arithmetic expressions at certain points with least significant bit accuracy. Applying this to (9.1) and (9.2) gives a significant improvement.

In [6] Böhm gave a large number of different algorithms for the inclusion of zeros of polynomials with automatic verification of correctness. Their presentation lies outside the scope of this article; we give only a few keywords. For a complete discussion cf. [6].

In [6] algorithms of higher order using higher order derivatives are given. Here the possibility is demonstrated of computing inclusions of the coefficients of quadratic factors of a polynomial. If, e.g., $p(x) = \sum_{i=0}^{n} a_i x^i$ is given, then $Ax^2 + Bx + C$, where $A,B,C\in\mathbb{IS}$ resp. $\mathbb{I}\mathbb{C}S$, is computed and the following is true. There exist $a,b,c\in\mathbb{R}$ resp. $\mathbb{C}$ with $a\in A$, $b\in B$, $c\in C$ such that $ax^2 + bx + c$ divides $p(x)$ without reminder. In this manner double zeros and, when including factors of higher degree, multiple zeros of a polynomial can be included. However, it cannot be verified that p has a zero of multiplicity greater than one.

Moreover in [6] several theorems and corresponding algorithms are derived using the Frobenius matrix of p and the transposed Frobenius matrix.

Next we briefly describe two methods for simultaneous inclusion of all complex zeros of a complex polynomial. The first one is an extension of a well-known procedure.

Theorem 9.3: (Böhm): Let $p(z) = \sum_{i=0}^{n} a_i \cdot z^i$ with $a_i\in\mathbb{C}S$ for $0\le i\le n$ and let $\tilde{z} = (\tilde{z}_1,\ldots,\tilde{z}_n)\in V\mathbb{C}S$ with $\tilde{z}_1\neq\tilde{z}_j$ for $1\le i,j\le n$ and $i\neq j$. If then for some $Z = (Z_1,\ldots,Z_n)\in\mathbb{I}V\mathbb{C}S$

$$\tilde{z}_1 \diamondsuit \diamondsuit_{(p)}(\tilde{z}_i) \diamondsuit \{a_n \diamondsuit \prod_{\substack{j=1\\ j\neq i}}^{n} (\tilde{z}_i \diamondsuit Z_j) \subseteq Z_i \quad \text{for } 1\le i\le n,$$

then for the zeros ξ_i, $1\le i\le n$ of p and a suitable indexing, we have $\xi_i\in\tilde{z}_i$ $\diamondsuit$ Z_i, $1\le i\le n$.

Here $\diamondsuit$ is defined as above. Notice, that p' is not needed.

The next theorem gives an improvement of the well-known method of Gargantini and Henrici for simultaneous inclusion of the complex zeros of complex polynomials.

Theorem 9.4: (Böhm): Let $p(z) = \sum_{i=0}^{n} a_i z^i$ with $a_i \in \mathbb{C}$ for $0 \leq i \leq n$ and let $\tilde{z} = (\tilde{z}_1,...,\tilde{z}_n) \in V\mathbb{C}$. Define $f: V\mathbb{C} \to V\mathbb{C}$, $f = (f_1,...,f_n)$ componentwise for $z = (z_1,...,z_n) \in \mathbb{C}$ by

$$f_i(z) = \tilde{z}_i - p(\tilde{z}_i)/\{p'(\tilde{z}_i) + p(\tilde{z}_i) \cdot \sum_{\substack{j=1 \\ j \neq 1}}^{n} (\tilde{z}_i - z_j)^{-1}\}, \quad 1 \leq i \leq n. \tag{9.3}$$

If then for some $Z \in \mathbb{IV}\mathbb{C}$, $Z = (Z_1,...,Z_n)$ the demoninator of (9.3) does not vanish for $z \in Z$, $1 \leq i \leq n$ and

$$\{f(z) \mid z \in Z\} \subsetneq Z, \tag{9.4}$$

then the zeros ξ_i, $1 \leq i \leq n$ of p satisfy $\xi_i \in \tilde{z}_i + Z_i$ with suitable indexing. Moreover for every $k \in \mathbb{N}$

$$\xi = (\xi_1,...,\xi_n) \in \{\tilde{z} + f_k(z) \mid z \in Z\}.$$

For the proof of the two preceding theorems cf. [6]. In contrast to the algorithm of Gargantini and Henrici an algorithm based on theorem 9.4 does not require inclusions for the zeros of p as an input. Moreover any complex arithmetic (rectangle, circular) can be used as long as the intervals are convex. The $\tilde{z}_i$ need not to be the midpoints of Z_i, in fact $\tilde{z}_i$ is not required to be an element of Z_i, $1 \leq i \leq n$.

Again the coefficients of the given polynomial may be intervals themselves. In this case the zeros of every point polynomial included by the interval polynomial are included.

The computing time for the new algorithms is of the same order as comparable (purely) floating-point algorithms. The latter, of course, offers none of the new features.

In the following we give some numerical examples. The algorithms are programmed on the UNIVAC 1108 of the University at Karlsruhe. There the floating-point screen is $S(2,27,-127,128)$. In fact the computational results of the algorithms derived from theorems 9.1, 9.2, 9.3 and 9.4 are almost identical, so we display the results for the last case only. The polynomials treated are:

P_1 with zeros $\pm\sqrt{2}$, 17/12, 41/29

P_2 with zeros $\pm\sqrt{2}$, 3363/2378

R_1 product of linear factors with random zeros in $[-2,2]$

R_2 coefficients randomly generated in $[-1,1]$

W $(x-1)(x-2)\cdot\ldots\cdot(x-11)-1$

L Legendre polynomial, coefficients computed in floating-point

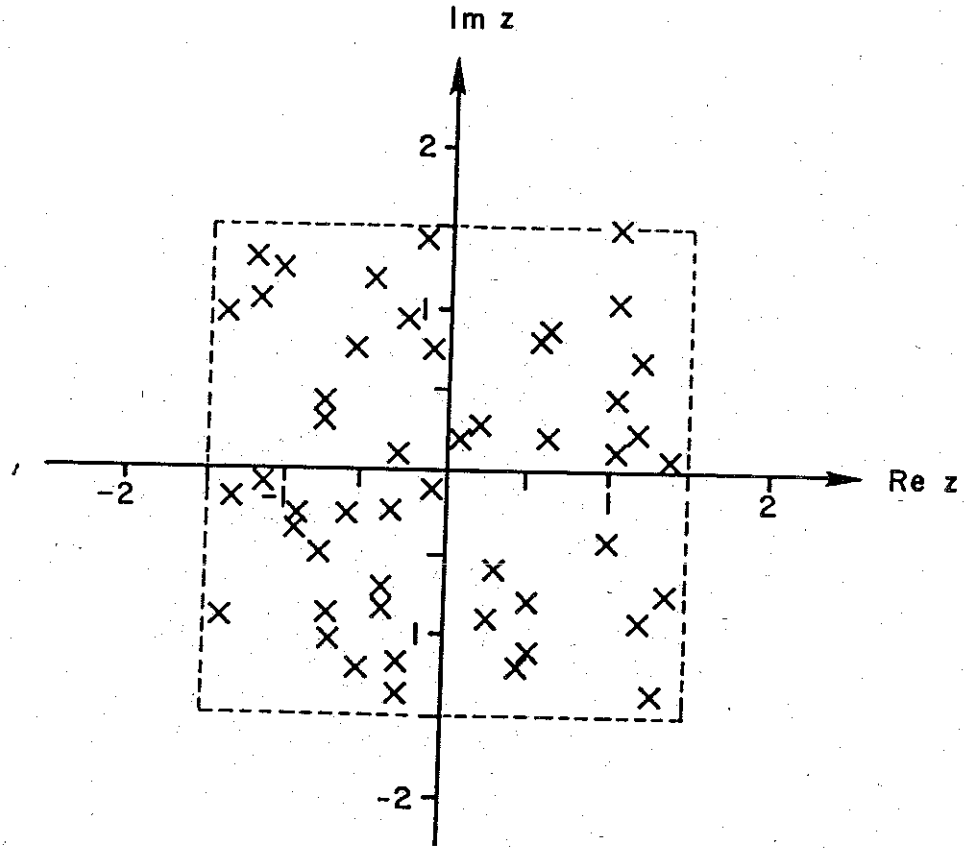
RC_1 randomly generated coefficients in the unit square

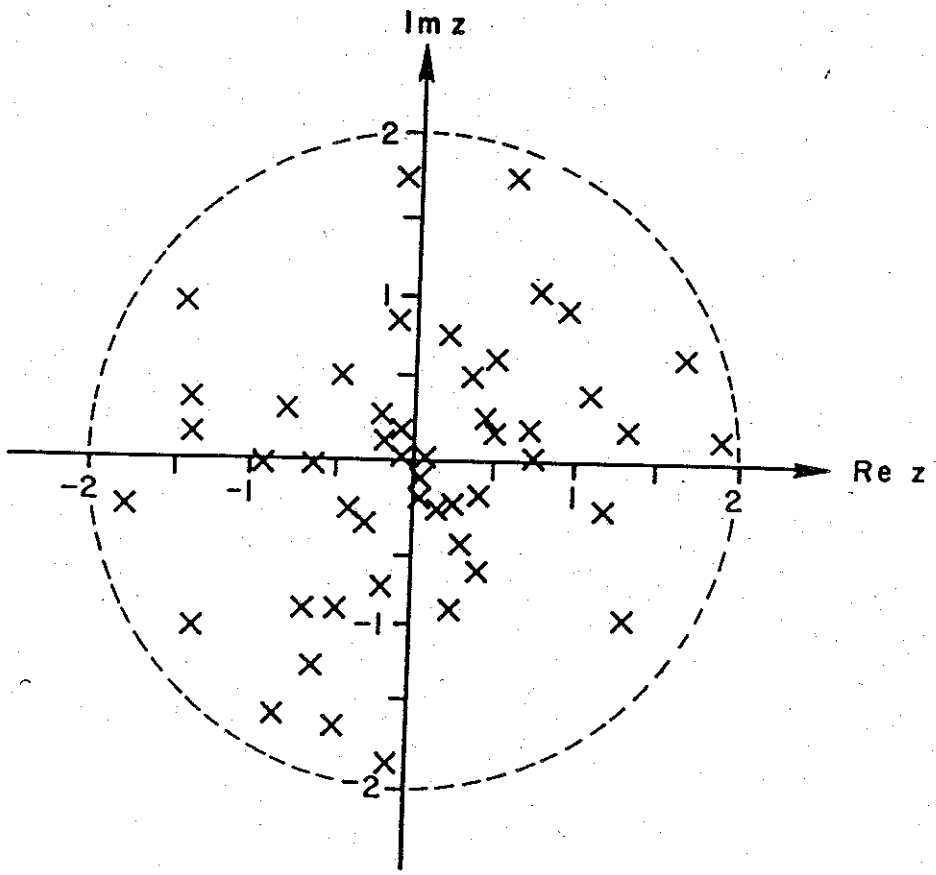
RC_2 randomly generated zeros with $|Re\, z| \leq 1.5$, $|Im\, z| \leq 1.5$

RC_3 products of linear factors, randomly generated with zeros in $|z| \leq 2$.

As an example we give two figures displaying the zeros of RC_2 and RC_3, both for degree 49.

The zeros of R_1 are particularly ill-conditioned. A procedure from IMSL implemented on the UNIVAC 1108 generated for R_1 of degree 50 an approximation -2.1 for a real zero, whereas -1.06 is the smallest real zero. The results are displayed in the following table. For more examples see [6].


Im z
2
1
-2
-1
1
2
-2
Re z


Im z
2
1
-2
-1
1
2
-1
-2
Re z

polynomial	degree	k	# of digits guaranteed
P_1	4	2	$8\frac{1}{2}$ (l.s.b.a.)
P_2	3	2	$8\frac{1}{2}$ (l.s.b.a.)
R_1	25	2	$8\frac{1}{2}$ (l.s.b.a.)
	50	3	8
R_2	25	2	$8\frac{1}{2}$ (l.s.b.a.)
	50	2	$8\frac{1}{2}$ (l.s.b.a.)
W	11	2	$8\frac{1}{2}$ (l.s.b.a.)
L	15	2	$8\frac{1}{2}$ (l.s.b.a.)
RC_1	25	2	$8\frac{1}{2}$ (l.s.b.a.)
	49	2	$8\frac{1}{2}$ (l.s.b.a.)
RC_2	25	2	$8\frac{1}{2}$ (l.s.b.a.)
	49	2	$8\frac{1}{2}$ (l.s.b.a.)
RC_3	25	2	$8\frac{1}{2}$ (l.s.b.a.)
	49	2	$8\frac{1}{2}$ (l.s.b.a.)

The initial X resp. Z is an interval with left and right bound equal to a floating-point approximation. Therefore (9.4) cannot be satisfied for the initial X resp. Z and an iteration is started similar to those described in chapter 7. In the table, k is the number of iterations, $k \geq 2$. In the last column the minimum number of digits guaranteed of all inclusions of all zeros is displayed. An additional l.s.b.a. means that the left and right bounds of all inclusions were consecutive points in the floating-point screen. The zeros of P_2 are

$$\pm\ 1.414213562... = \pm\sqrt{2}$$

and $$+1.414213625... = 3363/2378$$

Therefore between $+\sqrt{2}$ and 3363/2378 are only 4 points of the floating-point screen. Nevertheless all zeros have been included to least significant bit accuracy with automatic verification of the correctness.

10. LINEAR, QUADRATIC AND CONVEX PROGRAMMING

In this chapter our aim is to give algorithms which verify the optimality of a solution to a linear, quadratic or convex programming problem. We begin the discussion with linear programming problems. We use the same notation as in [9]. In this chapter all vectors are by definition column vectors, the transposed vector is indicated by a prime.

Let $x,p \in V_n\mathbb{R}$; $b \in V_m\mathbb{R}$, $A \in M_{m,n}\mathbb{R}$ and Q: $V_n\mathbb{R} \to \mathbb{R}$ where $Q(x) := p'x$. We suppose $m<n$. The problem is to find a vector $\hat{x} \in V_n\mathbb{R}$ with $\hat{x} \geq 0$ satisfying the condition $A\hat{x} = b$ and having the property

$$y \in V_n\mathbb{R},\ y \geq 0 \text{ and } Ay = b \Rightarrow Q(y) \geq Q(x).$$

We write this linear programming problem as follows:

$$Ax = b,\ x \geq 0 \text{ and } Q(x) = p'x = \text{Min!} \tag{10.1}$$

Let A_k, $1 \leq k \leq n$ be the column vectors of A and let Z be a set of indices in the range $1 \ldots n$. We suppose $|Z| = m$ and define $\tilde{A} \in M_{m,m}\mathbb{R}$ to be the matrix with columns A_k, $k \in Z$ and $\tilde{p} \in V_m\mathbb{R}$ to be the vector with components p_k, $k \in Z$. Let $\tilde{A}$ be non-singular and define $t' = (t_i) = \tilde{p}'\tilde{A}^{-1}A$. Define the vector $x^0 \in V_n\mathbb{R}$ in the following way: $x_j^0 := 0$ for $j \notin Z$ and the m components x_j^0, $j \in Z$ are the components of $\tilde{A}^{-1}b$ in successive order. Then the following is true (cf. [9]):

1) If $t_j \leq p_j$ for every $j \notin Z$, then x^0 is an optimal solution to (10.1).

2) If there is a $j \notin Z$ such that $t_j > p_j$ and $(\tilde{A}^{-1}A_j)_k \leq 0$ for $1 \leq k \leq m$, then (10.1) has no solution.

3) If $t_j > p_j$ for a $j \notin Z$ and a component of $\tilde{A}^{-1}A_j$ is greater than zero, then there is a feasible solution $x^1 \in V_n\mathbb{R}$ of (10.1) satisfying $Q(x^1) < Q(x^0)$.

Consider the linear systems

$$\begin{pmatrix} \tilde{A} & 0 \\ -p' & 1 \end{pmatrix} \cdot \begin{pmatrix} y_j \\ c_j \end{pmatrix} = \begin{pmatrix} A_j \\ -p_j \end{pmatrix}, \text{ where } y_j \in V_n\mathbb{R} \text{ and } c_j \in \mathbb{R}, j \notin Z. \tag{10.2}$$

Then (provided $\tilde{A}^{-1}$ exists)

$$y_j = \tilde{A}^{-1} \cdot A_j \text{ and } c_j = \tilde{p}'\tilde{A}^{-1}A_j - p_j = t_j - p_j, \quad j \notin Z.$$

So if $c_j \leq 0$ for all $j \notin Z$, then $\hat{x} \in V_n\mathbb{R}$ defined such that $\hat{x}_j := 0$ for $j \notin Z$ and the m components $\hat{x}_j$, $j \in Z$ are the m components of $\tilde{A}^{-1}b$, is an optimal solution to (10.1) (cf. [9]). This leads to an algorithmic application on computers with automatic verification of correctness. For this purpose the problem is formulated as follows:

$$\begin{aligned} &A \in M_{m,n}S;\ x,p \in V_nS;\ b \in V_mS \text{ and define } Q\colon V_n\mathbb{R} \to \mathbb{R} \text{ by } Q(x) := p'x. \\ &\text{Find an } \hat{x} \in V_n\mathbb{R} \text{ with} Q(\hat{x}) = \text{Min}\,! \text{ under the restrictions } \hat{x} \geq 0,\ A\hat{x} = b. \end{aligned} \tag{10.3}$$

Theorem 10.1: Let the linear programming problem (10.3) be given. Suppose inclusions $Y_j \in \mathbb{I}V_mS$ for y_j, resp. $C_j \in \mathbb{I}S$ for c_j, $j \notin Z$ in (10.2) using algorithm 2.1 have been computed. Then A is non-singular and the following is true:

1) If $\sup(C_j) \leq 0$ for every $j \notin Z$, then the vector $\hat{x} \in V_n\mathbb{R}$ defined such that $\hat{x}_j := 0$ for $j \notin Z$ and the m components $\hat{x}_j$, $j \in Z$ are the m components of $\tilde{A}^{-1}b$, is an optimal solution to (10.3).

2) If there is a $j \notin Z$ such that $\inf(C_j) > 0$ and $(\sup(Y_j))_k \leq 0$ for $1 \leq k \leq m$, then (10.3) has no solution.

3) If $\inf(C_j > 0$ for a $j \notin Z$ and $(\inf(Y_j))_k > 0$ for some $1 \leq k \leq m$, then a base vector $A_j, j \in Z$ has to be exchanged.

Proof: This is a consequence of corollary 2 and the preceding discussion. □

The linear system (10.2) always involves the same matrix for all right hand sides, so the total computing time reduces to $m^3 + 2(n-m)m^2$. The optimal solution can be included by setting the right hand side of (10.2) equal to $(b,0)$. Moreover the last component of this solution vector includes the optimal value $Q(\hat{x})$. Of course, the restriction $m < n$ can be omitted and dual problems can be treated in a similar way.

Next we discuss convex programming problems. Again we use the same notation as in [9]. Consider F: $V_n\mathbb{R}\to\mathbb{R}$, f_j: $V_n\mathbb{R}\to\mathbb{R}$ for $1\le j\le m$. Suppose F, f_j for $1\le j\le m$ to be convex, having first partial derivate. Then for $x\in V_n\mathbb{R}$

$$F(x) = \text{Min}\,!\ \text{with the restrictions } x\ge 0 \text{ and } f_j(x)\le 0,\ 1\le j\le m \tag{10.4}$$

is a convex programming problem. We define the Lagrange function $\phi(x,u)$: $V_{n+m}\mathbb{R}\to\mathbb{R}$, $x\in V_n\mathbb{R}$, $u\in V_m\mathbb{R}$ by (cf. [9])

$$\phi(x,u) := F(x) + u'\bullet f(x) \text{ with } f := (f_1,\ldots,f_m). \tag{10.5}$$

If ϕ_x resp. ϕ_u denotes the gradient of ϕ with respect to x resp. u:

$$\phi_x = \left(\frac{\partial\phi}{\partial x_1},\ldots,\frac{\partial\phi}{\partial x_n}\right)',\quad \phi_u = \left(\frac{\partial\phi}{\partial u_1},\ldots,\frac{\partial\phi}{\partial u_n}\right)', \tag{10.6}$$

then the following is true (Kuhn-Tucker, cf. [9]):

If there exists a feasible point $\bar{x}\in V_n\mathbb{R}$ with $f(\bar{x})<0$,
then $\hat{x}\in V_n\mathbb{R}$ with $\hat{x}\ge 0$ is an optimal solution of (10.4) (10.7)
if and only if there is a $u\in V_m\mathbb{R}$ with $u\ge 0$ satisfying

$$\begin{aligned} &\phi_x(\hat{x},u)\ge 0 \qquad && \hat{x}'\bullet\phi_x(\hat{x},u) = 0\\ &\phi_u(\hat{x},u)\le 0 && u'\bullet\phi_u(\hat{x},u) = 0. \end{aligned} \tag{10.8}$$

Since $x,u,>0$ condition (10.8) is equivalent to

$$\begin{aligned} &\phi_x(\hat{x},u)\ge 0, \qquad && \phi_u(\hat{x},u)\le 0\\ &\text{for every } 1\le j\le n && \hat{x}_j \text{ or } (\phi_x(\hat{x},u))_j \text{ equals zero}\\ &\text{for every } 1\le j\le m && u_j \text{ or } (\phi_u(\hat{x},u))_j \text{ equals zero.} \end{aligned} \tag{10.9}$$

Suppose the convex programming problem (10.4) is to be solved on a computer including automatic verification of correctness of the result. Assume there are functions $\diamondsuit f$: $V_n\mathbb{R}\to\mathbb{IV}_nS$, $\diamondsuit F$: $V_n\mathbb{R}\to\mathbb{IS}$, $\diamondsuit\phi_x$: $V_{n+m}\mathbb{R}\to\mathbb{IV}_nS$ and $\diamondsuit\phi_u$: $V_{n+m}\mathbb{R}\to\mathbb{IV}_mS$ satisfying

$$x\in V_n\mathbb{R} \Rightarrow f(x)\in \diamondsuit(x) \text{ and } F(x)\in \diamondsuit(x)$$
$$(x,u)\in V_{n+m}\mathbb{R} \Rightarrow \phi_x(x,u)\in \diamondsuit_{\phi_x}(x,u) \text{ and } \phi_u(x,u)\in \diamondsuit_{\phi_u}(x,u). \tag{10.10}$$

Let $\bar{x}\in V_nS$ with $\bar{x}\geq 0$ and $\sup(\diamondsuit(\bar{x}))<0$ be given and let floating-point approximations $\tilde{x}\in V_nS$ and $\tilde{u}\in V_mS$ for a solution of (10.8) be given. Let I resp. J be the set of indices i resp. j for which $\tilde{x}_i$ resp. $\tilde{u}_j$ is approximately zero. Consider the following system of non-linear equations.

$$\begin{aligned}\phi_{x_i}(x,u)\cdot x_i &= 0 \quad \text{for } 1\leq i\leq n \text{ and } i\notin I\\ \phi_{u_j}(x,u)\cdot u_j &= 0 \quad \text{for } 1\leq j\leq m \text{ and } j\notin J.\end{aligned} \tag{10.11}$$

These are $n+m-|I|-|J|$ equations in the same number of unknowns when omitting x_i for $i\in I$, u_j for $j\in J$ in the computation of $\phi_{x_i}(x,u)$, $\phi_{u_j}(x,u)$. To this non-linear system algorithm 7.1 is applicable.

Theorem 10.2: Solve the non-linear system (10.11) using Algorithm 7.1 and let X_i, $1\leq i\leq n$, $i\notin I$ and U_j, $1\leq j\leq m$, $j\notin J$ be the computed inclusions for the solutions. Define $X_i := 0$ for $i\in I$ and $U_j := 0$ for $j\in J$ and let $X := (X_1,\ldots,X_n)\in \mathbb{I}V_nS$ and $U := (U_1,\ldots,U_m)\in \mathbb{I}V_mS$. If then

$$\begin{aligned}&\inf(X_i)\geq 0 \text{ and } \inf(U_j)\geq 0 \text{ for } 1\leq i\leq n,\ 1\leq j\leq m \text{ and}\\ &\inf\{\diamondsuit_{\phi_x}(X,U)\}\geq 0 \text{ and } \sup\{\diamondsuit_{\phi_u}(X,U)\}\leq 0,\end{aligned}$$

then the convex programming problem (10.4) has an optimal solution $\hat{x}\in X$.

Next we discuss quadratic programming problems. Again we use the same notation as in [9]. To solve a quadratic programming problem with automatic verification of correctness of the result, theorem 10.2 could be used. However, taking advantage of the special structure of the problem finally leads to a system of linear equations as will be shown now.

Let $A\in M_{m,n}\mathbb{R}$; $x,p\in V_n\mathbb{R}$, $b\in V_m\mathbb{R}$ and let $C\in M_{n,n}\mathbb{R}$ be a symmetric, positive definite matrix. Then a quadratic programming problem is given by (cf. [9]):

$$Q\colon V_n\mathbb{R}\to\mathbb{R} \text{ with } Q(x) := p'x + x'Cx = \text{Min}! \text{ with } Ax\leq b,\ x\geq 0. \tag{10.12}$$

A specialization of the Kuhn-Tucker Theorem yields (cf. [9]):

A vector $\hat{\hat{x}} \in V_n\mathbb{R}$ with $\hat{\hat{x}} \geq 0$ is an optimal solution to (10.12) if and only if there exist $u \in V_m\mathbb{R}$, $v \in V_n\mathbb{R}$, and $y \in V_m\mathbb{R}$ such that

$$\begin{aligned} &A\hat{\hat{x}} + y = b,\ v - 2Cx - A'u = p \\ &u \geq 0,\ v \geq 0,\ y \geq 0 \end{aligned} \tag{10.13}$$

and

$$xv + yu' = 0. \tag{10.14}$$

The assumption of the existence of an $\bar{x} \in V_n\mathbb{R}$ with $f(\bar{x}) < 0$, as in the convex case, can be omitted because the restrictions are affine-linear. Condition (10.14) means because of $x,v,y,u \geq 0$ that for $1 \leq i \leq n$ resp. $i \leq j \leq m$ either $x_i = 0$ or $v_i = 0$ resp. either $y_j = 0$ or $u_j = 0$. Thus we can proceed in the following way.

Consider the system of non-linear equations

$$Ax + y = b,\ v - 2Cx - A'u = p, x_i v_i = 0 \text{ for } 1 \leq i \leq n,\ y_j u_j = 0 \text{ for } 1 \leq j \leq m \tag{10.15}$$

in $2n + 2m$ unknowns (x,v,y,u). Let $(\tilde{x},\tilde{v},\tilde{y},\tilde{u})$ be an approximate solution of (10.15). It follows from (10.15) that for every $1 \leq i \leq n$ either x_i or v_i equals zero and for every $1 \leq j \leq m$ either y_j or u_j equals zero. Delete in (10.15) every variable x_i,v_i,y_j,u_j for which $\tilde{x}_i,\tilde{v}_i,\tilde{y}_j,\tilde{u}_j$ is approximately zero. Then $n + m$ equations

$$Ax^* + y^* = b \text{ and } v^* - 2Cx^* - A'u^* = p \tag{10.16}$$

remain, where in x^*,v^*,y^*,u^* have on the whole $n + m$ fewer components than x,v,y,u. The system (10.16) is linear.

Theorem 10.3: Let $A \in M_{m,n}S$, $p \in V_nS$, $b \in V_mS$ and let $C \in M_{n,n}$ be a symmetric, positive definite matrix. Define Q: $V_n\mathbb{R} \to \mathbb{R}$ by

$$x \in V_n\mathbb{R}: Q(x) := p'x + x'Cx.$$

If the linear system (10.16) has been solved using algorithm 2.1 with including intervals

$X^*, V^* \in \mathbb{IV}_n S$ and $Y^*, U^* \in \mathbb{IV}_m S$ of the solution, then the following is true:

> If $\inf(X^*) \geq 0$, $\inf(V^*) \geq 0$, $\inf(Y^*) \geq 0$ and $\inf(U^*) \geq 0$, then the quadratic programming problem (10.12) has an optimal solution $\hat{x} \in V_n\mathbb{R}$. The non-zero components of $\hat{x}$ are included in X^*, the others are zero respective to the procedure generating (10.16) described above.

the presented theorems 10.1, 10.2 and 10.3 lead to algorithms for automatic verification of the optimality of an approximate solution to a linear, convex and quadratic programming problem as in chapters 2 and 7. The presented theorems and the corresponding algorithms can easily be extended to uncertain data. They are similarly applicable to dual optimization problems.

Computational results of the corresponding algorithms are for instance those presented in chapters 2 and 7 of this article for linear and non-linear problems.

11. ARITHMETIC EXPRESSIONS

Single precision floating-point computations may yield an arbitrarily false result due to cancellation and rounding errors. This is true even for very simple, structured expressions such as Horner's scheme for polynomial evaluation. A simple procedure will be presented for fast calculation of the value of an arithmetic expression to least significant bit accuracy in single-precision computation. For this purpose in addition to the usual floating-point arithmetic, only a precise scalar product is required. If the approximation computed by usual floating-point arithmetic is good enough, the computing time for the new algorithm is approximately the same as for usual floating-point computation. If not, the essential advantage of the algorithm presented here is that the inaccuracy of the approximation is recognized and corrected. An inclusion with least significant bit accuracy for the value of the arithmetic expression is computed with automatic verification of correctness. Following we give a brief description of the procedure. For more details cf. [5], [33].

Let S be the floating-point screen of the computer in use. Then elements of S are named constants. Arithmetic expressions consist of constants and $x, -, \bullet, /, (\ , \ )$. An arithmetic expression can be transformed to the quotient of two arithmetic expressions where in the

numerator and denominator quotients may occur but only with constants in the denominator. Further an expression can be altered in such a way that in every product at most one factor is an expression itself, the others are constants. Example:

$$a^2-b+\frac{4a^2}{b(b-a)} \rightarrow \frac{(a^2-b)b-(a^2-b)a+4a^2/b}{b-a}. \tag{11.1}$$

This process can be performed automatically (cf. [8]). Therefore we consider arithmetic expressions which can be obtained by applying the following rules:

1) A constant is an expression.

2) The sum and difference of two expressions is an expression.

3) The product of an expression and a constant is an expression.

4) An expression divided by a constant is an expression.

Such expressions are called simple (arithmetic) expressions.

When evaluating a simple expression, each rule 1...4 corresponds to the evaluation of an intermediate result. Let a,b,c be constants and x,y be values of subterms. Then a new intermediate result z is obtained in one of the following ways:

1) $z=a$

2) $z=x\pm y$

3) $z=x\cdot a$

4) $z=x/a$ or $a\cdot z=x$.

Thus a simple arithmetic expression can be regarded as a system of linear equations. The variables are the intermediate results. Applying only rules 1), 2), 3) and 4) may result in many variables. The number of variables can be reduced. Example for (11.1):

$$\begin{aligned} x_1&=a & x_4&=x_3\cdot b\\ x_2&=x_1\cdot a & b\cdot x_5&=x_2\\ x_3&=x_2-b & x_6&=-a\cdot x_3+x_4+4x_5 \end{aligned} \tag{11.2}$$

For calculating the value of a polynomial $p(\xi) = \sum_{i=0}^{n} a_i \bullet \xi^{n-i}$ we obtain the linear system

$$x_0 = a_0;\ x_{i+1} = \xi \bullet x_i + a_{i+1} \text{ for } 0 \le i \le n-1.$$

Obviously the linear system corresponding to a simple arithmetic expression is lower triangular. This it can be solved by forward substitution and, which is important, this process can be iterated. Moreover not only approximations are achieved but also an inclusion of the value of the arithmetic expression with automatic verification of correctness.

The following remarks hold for arbitrary lower triangular linear systems and especially for linear systems corresponding to simple arithmetic expressions.

Let $L \in MS$, $b \in VS$ with $L_{ij} = 0$ for $i<j$, $1 \le i, j \le n$ and consider

$$Lx = b \quad \text{with} \quad L_{ii} \neq 0,\ 1 \le i \le n. \tag{11.3}$$

Then (11.3) can be solved using Bohlender's algorithm (cf. [3]) by

$$\hat{x}_k \in X_k, \quad 1 \le k \le n. \tag{11.4}$$

This process can be iterated using the residue iteration. The corresponding algorithm is given in [33]. Let X^k, $k \ge 1$ denote the computed inclusion vector ($X^k \in \mathbb{I}VS$) for $\hat{x}$ after the k^{-th} iteration step. Then in [33] the following is proved.

Lemma 11.1: Let ε be the unit of relative rounding error. If then no over- or underflow occurred during computation then for some norm $\| \bullet \|$

$$\| d(X^{k+1}) \| \le \varepsilon \bullet \eta \bullet \| d(X^k) \| \quad \text{with} \quad \eta = 5n^2 + 0(\varepsilon).$$

The constant η is given explicitly in [33]. Lemma 11.1 covers *all* rounding errors due to arithmetic operations in the floating-point screen. If ℓ is the length of the mantissa of the computer in use then, by the preceding lemma, the diameter of the inclusion improves in every iteration step by a factor of at least $5n^2 \bullet B^{-\ell+1}$, where B is the base of the floating-point screen in use.

There are direct extensions to arithmetic expressions consisting of complex numbers. The result is a complex interval with least significant bit accuracy. The arithmetic may be rectangular, circular, segment or any other.

In the following we give some computational results. For more examples see [5], [33].

1) $y^2(4x^4+y^2-4x^2)-8x^6$ for $x=470832$, $y=665857$

2) $\sum_{i=1}^{5} x_i^2-\frac{1}{5}\cdot(\sum_{i=1}^{5} x_i)^2$ for $x_i=7.951_{10}7+i-3$, $i=1(1)5$.

Remark: Expressions like this occur in least square approximation.

3) $f(x)=((543339720x-768398401)x-1086679440)x+1536796802$

a) $x=1.4142$

b) $x=1.41421356238$

c) $x=1.414213561$

4) $(f(x-h)-2f(x)+f(x+h))/h^2$ with $f(x)=\dfrac{2734x-2761}{4556x^2-9247x+4692}$ for $x=1$.

This is an approximation for $f''(1)$.

5) $\sum_{i=0}^{90}(-1)^i\cdot\frac{x^i}{i!}$ for $x=20$.

This is an approximation to e^{-20}.

The following table shows computational results computed on a minicomputer based on Z80 with floating-point screen (10,12, – 99,99). In the columns of the table are displayed from left to right:

- The number of the example
- The floating-point approximation $\tilde{x}$
- The correct value $\hat{x}$ of the expression rounded to 12 decimal digits

- The number k of iterations
- The final result X of the new algorithm.

	$\tilde{x}$	$\hat{x}$	k	X
1)	$+5.0_{10}23$	$+1.0$	2	$+1.0$
2)	-100000.0	$+10.0$	1	$+10.0$
3a)	$+0.2800$	$+0.282673919360$	1	$+0.282673919360$
3b)	$+0.01$	$+7.32719247117_{10}{-14}$	2	$+7.3271924711^{8}_{710}{-14}$
3c)	-0.01	$+2.89746134369_{10}{-9}$	2	$+2.8974613436^{9}_{810}{-9}$

Example 4) resp. 5) are approximations to $f''(1)$ resp. e^{-20}. It seems not to be meaningful to give the exact value of an approximation. Therefore in the following table we display only the leading digits of $\tilde{x}$ and X to demonstrate their discrepancy. However, all inclusions X were computed with least significant bit accuracy. The exact value of $f''(1)$ in example 4) is 54. Notice, that the final summand in Example 5) is $20^{90}/90!$

Example		$\tilde{x}$	X
4)	$h=10^{-1}$	5645	5645
	$h=10^{-2}$	3788500	378500
	$h=10^{-3}$	1184	1185
	$h=10^{-4}$	125.0	65.21
	$h=10^{-5}$	4900	54.11
	$h=10^{-6}$	380000	54.001
	$h=10^{-7}$	-10000000	54.00001
	$h=10^{-12}$	0	54.0000000002, 53.9999999999
5)		$1.188_{10}{-4}$	$2.0611536224^{4}_{310}{-9}$

The initial floating-point approximation is the result when evaluating the expression using usual floating-point arithmetic. Either, the final result with automatic verification of correctness is a point interval or else (in examples 3b, 3c) identical digits of the left and right bound are displayed only once.

If the initial approximation is "good enough", one iteration is executed to achieve least significant bit accuracy. In this case the total computing time is of the same order as usual floating-point evaluation.

The algorithm for computing the value of an arithmetic expression to least significant bit accuracy gives a significant improvement of the algorithm for non-linear systems presented in chapter 7 (applicable to those functions not consisting of transcendental functions) and of the algorithms for including real or complex zeros of a polynomial presented in chapter 9. It is obvious that if (in the latter case) the value of a polynomial is not correct computable, then arbitrarily false results may be computed. Consider the following example:

$$P(x) = 67872320568x^3 - 95985956257x^2 - 135744641136x + \\ + 191971912515.$$

Newton's procedure was applied to P with starting point $x = 2$, where $P(x)$ and $P'(x)$ were evaluated using Horner's scheme and usual floating-point arithmetic. The floating-point screen in use is (10,12, – 99,99) in which the coefficients of P and P' are storable without rounding error. The arithmetic in use satisfies (R), (R1), (R2), (R4) and (R6) given in chapter 1.

In the following table in the left column (cf. [37]) the

x^k	$x^{k+1}-x^k$
1.73024785661	$2.698_{10}{-01}$
1.57979152125	$1.505_{10}{-01}$
1.49923019011	$8.056_{10}{-02}$
1.45733317058	$4.190_{10}{-02}$
1.43593403289	$2.140_{10}{-02}$
1.42511502231	$1.082_{10}{-02}$
1.41967473598	$5.440_{10}{-03}$
1.41694677731	$2.728_{10}{-03}$
1.41558082832	$1.366_{10}{-03}$
1.41489735833	$6.835_{10}{-04}$
1.41455549913	$3.419_{10}{-04}$
1.41438453509	$1.710_{10}{-04}$
1.41429903606	$8.550_{10}{-05}$
1.41425628589	$4.275_{10}{-05}$
1.41423488841	$2.140_{10}{-05}$
1.41422414110	$1.075_{10}{-05}$
1.41421847839	$5.663_{10}{-06}$
1.41421582935	$2.649_{10}{-06}$
1.41421353154	$2.298_{10}{-06}$
1.41421353154	0
1.41421353154	0
1.41421353154	0
1.41421353154	0
1.41421353154	0

iterates x^k are displayed and in the right column the difference $x^{k+1} - x^k$ of two successive iterates. The iteration is monotone, the difference of two successive iterates decreases and

$$\bar{x} = 1.41421353154$$

is a fixed point. Computation in $\mathbb{R}$ would imply $P(\bar{x}) = 0$. However, with the new algorithm to compute the value of an arithmetic expression to least significant bit accuracy we obtain

$$P(\bar{x}) \in 1.0001825038_1^2,$$

and in fact the smallest value for $P(x)$ for $x>0$ is approximately 1.

CONCLUSIONS

In the preceding chapters the theoretical background and corresponding algorithms have been given for several problems in numerical analysis. The algorithms for solving linear systems (dense, band, overdetermined, underdetermined and sparse), inversion of matrices and evaluation of arithmetic expressions compute an inclusion of the solution with automatic verification of correctness, existence and uniqueness; all this in a self-contained manner. The other algorithms for non-linear systems, algebraic eigenvalue problems, zeros of real and complex polynomials and linear, quadratic and convex programming problems provide an approximation of the solution for use in computing an inclusion of the solution with automatic verification of correctness, existence and uniqueness. This approximation can be obtained by any floating-point algorithm. Therefore the latter procedures estimate the error of an approximate solution. In other words they verify the correctness of an error margin (in addition to verification of existence and uniqueness). These "verification-algorithms" could replace additional tests such as altering input data, recomputing in higher precision etc. These tests would have to be developed and utilized for each individual problem by the programmer. The new algorithms perform the verification automatically without any effort on the part of the user, without any knowledge about the condition of the problem and, most importantly, without either a deep mathematical background or an extensive investigation. This, of course, is also true for the algorithms which compute an inclusion of the solution directly without initial approximation. The automatic error control is a key property of all the algorithms presented here.

The efficiency of the algorithms has been demonstrated by inverting the Hilbert 21×21 matrix on a 14 hexadecimal digit computer. This is (after multiplying with a proper factor) the Hilbert matrix of largest dimension which can be stored without rounding errors in this floating-point system. The error bounds for all components of the inverse of the Hilbert 21×21 matrix are as small as possible, i.e., left and right bounds differ only by one in the last place of the mantissa of each component. We call this least significant bit accuracy (1sba). Our experience shows that the results of the algorithms using our new methods very often have the 1sba-property for every component of the solution.

REFERENCES

[1] Abbott, J. P., Brent, R. P. (1975). Fast Local Convergence with Single and Multistep Methods for Nonlinear Equations, Austr. Math. Soc. 19 (series B), 173-199.

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

[3] Bohlender, G. (1977). Floating-point computation of functions with maximum accuracy. IEEE Trans. Comput. C-26, No. 7, 621-632.

[4] Bohlender, G., Grüner, K. Gesichtspunkte zur Implementierung einer optimalen Arithmetik. "Wissenschaftliches Rechnen und Programmiersprachen", Herausgeber U. Kulisch and Ch. Ullrich, B. G. Teubner Stuttgart.

[5] Böhm, H. Auswertung arithmetischer Ausdrücke mit maximaler Genauigkeit. "Wissenschaftliches Rechnen und Programminersprachen", Herausgeber U. Kulisch and Ch. Ullrich, B. G. Teubner Stuttgart.

[6] Böhm, H. (1980). Berechnung von Schranken für Polynomwurzeln mit dem Fixpunktsatz von Brouwer. Interner Bericht des Inst. f. Angew. Math., Universität Karlsruhe.

[7] Böhm, H. Private Communication.

[8] Böhm, H. (1981). Automatische Umwandlung eines arithmetischen Ausdrucks in eine zur exakten Auswertung geeignete Form. Interner Bericht des Inst. f. Angew. Math., Universität Karlsruhe.

[9] Collatz, L., Wetterling, W. (1966). Optimierungsaufgaben. Heidelberger Taschenbücher, Band 15, Springer-Verlag, Berlin-Heidelberg-New York.

[10] Collatz, L. (1968). *Funktionalanalysis und Numerische Mathematik,* Springer-Verlag.

[11] Forsythe, G. E., Moler, C. B. (1967). *Computer Solution of Linear Algebraic Systems,* Prentice-Hall.

[12] Forsythe, G. E. (1970). Pitfalls in computation, or why a Math book isn't enough, Technical Report No. CS147, Computer Science Department, Stanford University, 1-43.

[13] Gastinel, N. (1972). *Lineare Numerische Analysis.* F. Vieweg & Sohn, Braunschweig.

[14] Hansen, E. Interval Arithmetic in Matrix Computations, Part 1. SIAM J. Numer. Anal. 2, 308-320 (1965), Part II. SIAM J. Numer. Anal. 4, 1-9 (1967).

[15] Heuser, H. (1967). *Funktionalanalysis.* Mathematsche Leitfäden, B. G. Teubner, Stuttgart.

[16] Kaucher, E., Rump, S. M. (1980). Generalized iteration methods for bounds of the solution of fixed point operator equations, Computing 24, 131-137.

[17] Kaucher, E., Rump, S. M. (1982). E-methods for Fixed Point Equations $f(x) = x$, Computing 28, p. 31-42.
[18] Knuth, D. (1969). "The Art of Computer Programming", Vol. 2, Addison-Wesley, Reading, Massachusetts.
[19] Krawczyk, R. (1969). Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken, Computing, 4, 187-120.
[20] Köberl, D. (1980). The Solution of Non-linear Equations by the Computation of Fixed Points with a Modification of the Sandwich Method, Computing, 25, 175-178.
[21] Kulisch, U., Miranker, W. L. (1981). Computer Arithmetic in Theory and Practice. Academic Press, New York.
[22] Kulisch, U. (1969). Grundzüge der Intervallrechnung, Überblicke Mathematik 2, Herausgegeben von D. Laugwitz, Bibliographisches Institut, Mannheim, S. 51-98.
[23] Kulisch, U. An Axiomatic Approach to Rounded Computations, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, TS Report No. 1020, 1-29 (1969), and Numer. Math. 19, 1-17 (1971).
[24] Kulisch, U. (1976). Grundlagen des numerischen Rechnens (Reihe Informatik, 19). Mannheim-Wien-Zürich: Bibliographisches Institut.
[25] Kulisch, U., Wippermann, H.-W. PASCAL-SC, PASCAL für wissenschaftliches Rechnen, Gemeinschaftsentwicklung von Institut für Angewandte Mathematik, Universität Karlsruhe (Prof. Dr. U. Kulisch), Fachbereich Informatik, Universtät Kaiserslautern (Prof. Dr. H.-W. Wippermann).
[26] Martinez, J. M. (1980). Solving Non-linear Simultaneous Equations with a Generalization of Brent's Method, BIT, 20, 501-510.
[27] McShane, E. J., Botts, T. A. (1959). *Real Analysis.* Von Nostrand.
[28] Meinardus, G. (1964). *Approximation von Funktionen und ihre numerische Behandlung,* Berlin-Göttingen-Heidelberg-New York; Springer, 180 S.
[29] Moore, R. E. (1966). *Interval Analysis.* Prentice-Hall.
[30] Moore, R. E. (1977). A Test for Existence of Solutions for Non-Linear Systems, SIAM J. Numer. Anal., 4.
[31] Moré, J. J., Cosnard, M. Y. (1979). Numerical Solution of Non-Linear Equations. ACM Trans. on Math. Software, Vol. 5, No. 1, 64-85.
[32] Ortega, J. M., Reinboldt, W. C. (1970). *Iterative Solution of Non-linear Equations in several Variables.* Academic Press, New York-San Francisco-London.
[33] Rump, S. M., Böhm, H. Least Significant Bit Evaluation of Arithmetic Expressions in Single-precision, to appear in Computing.
[34] Rump, S. M. (1980). Kleine Fehlerschranken bei Matrixproblemen, Dr.-Dissertation, Inst. f. Angew. Math., Universität Karlsruhe.
[35] Rump, S. M. (1979). Polynomial Minimum Root Separation, Math. of Comp. Vol. 33, No. 145, 327-336.
[36] Rump, S. M. (1982). Solving Non-linear Systems with Least Significant Bit Accuracy, Computing 29, 183-200.
[37] Rump, S. M. Rechnervorführung, Pakete für Standardprobleme der Numerik, "Wissenschaftliches Rechnen und Programmiersprachen", Herausgeber U. Kulisch and Ch. Ullrich, B. G. Teubner Stuttgart.
[38] Rump, S. M. Lösung linearer und nichtlinearer Gleichungssysteme mit maximaler Genauigkeit. "Wissenschaftliches Rechnen und Programmiersprachen", Herausgeber U. Kulisch and Ch. Ullrich, B. G. Teubner Stuttgart.
[39] Schwarz, H. R., Rutischauser, H., Stiefel, E. (1972). *Matrizen-Numerik,* B. G. Teubner Stuttgart.
[40] Stoer, J. (1972). *Einführung in die Numerische Matematik I.* Heidelberger Taschenbücher, Band 105, Springer-Verlag, Berlin-Heidelberg-New York.

[41] Stoer, J., Bulirsch, R. (1973). *Einführung in die Numerische Mathematik II.* Heidelberger Taschenbücher, Band 114, Springer-Verlag, Berlin-Heidelberg-New York.
[42] Varga, R. S. (1962). *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, New Jersey.
[43] Wilkinson, J. H. (1969). *Rundungsfehler.* Springer-Verlag.
[44] Wongwises, P. Experimentelle Untersuchungen zur numerischen Auflösung von linearen Gleichungssystemen mit Fehlererfassung, Interner Bericht 75/1, Institüt für Praktische Mathematik, Universität Karlsruhe.
[45] Yohe, J. M. (1973). Interval Bounds for Square Roots and Cube Roots, Computing 11, 51-57.
[46] Yohe, J. M. (1973). Roundings in Floating-Point Arithmetic, IEEE Trans. on Comp., Vol. C12, No. 6, 577-586.
[47] Alefeld, G. (1979). Intervallanalytische Methoden bein nicht-linearen Gleichungen, In "Jahrbuch Überblicke Mathematik 1979", B. I. Verlag, Zürich.
[48] Rump, S. M., Kaucher, E. (1980). Small Bounds for the Solution of Systems of Linear Equations, Computing Suppl. 2.