Mathematical Foundation of Computer Arithmetic

ULRICH KULISCH

Abstract—**During recent years a number of papers concerning a mathematical foundation of computer arithmetic have been written. Some of these papers are still unpublished. The papers consider the spaces which occur in numerical computations on computers depending on a properly defined computer arithmetic. The following treatment gives a summary of the main ideas of these papers. Many of the proofs had to be sketched or completely omitted. In such cases the full information can be found in the references.**

Index Terms—**Axiomatic definition of computer arithmetic, floating-point arithmetic, interval arithmetic, numerical analysis, rounding analysis, theory and implementation of computer arithmetic.**

I. Introduction

NUMERICAL algorithms are usually derived and defined in one of the spaces R of real numbers, VR of vectors, or MR of matrices over the real numbers. Besides these spaces, the corresponding complex spaces C, VC, and MC also occur occasionally. Several years ago numerical analysts also began to define and study algorithms for intervals over these spaces. If we denote the set of intervals over an ordered set $\{M,\leqq\}$ by IM we get the spaces IR, IVR, IMR and IC, IVC, and IMC. See the second column in Fig. 1.

Since a real number in general is represented by an infinite b-adic expansion, the algorithms given in these spaces in general cannot be executed within them. The real numbers, therefore, get approximated by a subset T in which all operations are simple and rapidly performable. On computers for T a floating-point system with a finite number of digits in the mantissa is used. If the desired accuracy cannot be achieved by computations within T, a larger system S with the property $R \supset S \supset T$ is used. Over T, respectively S, we can now define vectors, matrices, intervals, and so on as well as the corresponding complexifications. Doing this we get the spaces VT, MT, IT, IVT, IMT, CT, VCT, MCT, ICT, $IVCT$, $IMCT$, and the corresponding spaces over S. See the third and fourth columns in Fig. 1. In the practical case of a computer, T and S can be understood as the sets of floating-point numbers of single and double length. In Fig. 1, however, S and T are only examples for a whole system of subsets of R with properties which will be defined later.

Now in every set of the third and fourth columns of Fig. 1, operations are to be defined. See the fifth column in Fig. 1. Furthermore the lines in Fig. 1 are not independent of each other. A vector can be multiplied by a number as well as by a matrix and an interval vector by an interval as well as by an interval matrix. In a good programming system the operations in the sets of the third and fourth columns in Fig. 1 should be available possibly as operators for special data types.

This paper is devoted to the question of how these operations are to be defined and in which structures they result. We shall see that all these operations can be defined by a simple, general, and common concept which allows us to describe all the sets listed in Fig. 1 by two abstract structures. More precisely, the structures derived from R can be described as ordered ringoids, respectively as ordered vectoids, while those derived from C are weakly ordered ringoids, respectively weakly ordered vectoids. (For definitions, see below.)

We are now going to describe this general principle in more detail. Let M be one of the sets listed in Fig. 1 and $\overline{M}$ a set of rules (axioms) given for the elements of M. Then we call the pair $\{M,\overline{M}\}$ a structure. In Fig. 1 the structure is well known in the sets of R, VR, MR, C, VC, and MC. Now let M be one of these sets and $*$ be one of the operations defined in M. Then also in the power set PM, which is the set of all subsets of M, an operation $*$ can be defined by

$$\bigwedge_{A,B\in PM} A * B := \{a * b \mid a \in A \wedge b \in B\}. \tag{1}$$

If we apply this definition for all operations $*$ of M we shall also see below that in the power set a structure $\{PM,\overline{PM}\}$ can be derived from that in $\{M,\overline{M}\}$. Summarizing this result we can say that in Fig. 1 the structure $\{M,\overline{M}\}$ is always known in the left-most element of every line. We are now looking for a general principle which allows us, beginning with the structure in the left-most element of every line, also to derive a structure in the subsets to the right-hand side.

First of all we define that the elements of a set M have to be transferred into the elements of a subset N on the right-hand side by a rounding. A mapping $\square: M \to N, N \subseteq M$, is called a "rounding" if it has the property

$$\text{(R1)} \quad \bigwedge_{a\in N} \square a = a.$$

Further, in all structures of Fig. 1 which we already know, a minus operator is defined and if, for instance, S and T are floating-point systems it is easy to see (see [11]–[14], [16], [19]) that in every line in Fig. 1 all subsets have the property

Manuscript received January 20, 1976; revised February 28, 1977.
The author is with the Institute of Applied Mathematics, University of Karlsruhe, Karlsruhe, Germany.

	R	S	T	+ - · / ×
	VR	VS	VT	+ - ×
	MR	MS	MT	+ - ·
PR	IR	IS	IT	+ - · / ×
PVR	IVR	IVS	IVT	+ - ×
PMR	IMR	IMS	IMT	+ - ·
	C	CS	CT	+ - · / ×
	VC	VCS	VCT	+ - ×
	MC	MCS	MCT	+ - ·
PC	IC	ICS	ICT	+ - · / ×
PVC	IVC	IVCS	IVCT	+ - ×
PMC	IMC	IMCS	IMCT	+ - ·

Fig. 1. Table of the spaces and operations occurring in numerical computations.

$$\text{(S)} \quad \bigwedge_{a \in N} -a \in N \wedge o,e \in N,$$

where o denotes the neutral element of addition and e the neutral element of multiplication if it exists.

It will turn out below that the rounding $\square: M \to N$ is responsible not only for the mapping of the elements but also for the resulting structure in the subsets N. If the structure $\{M,\overline{M}\}$ is given, the structure $\{N,\overline{N}\}$ is essentially dependent by the properties of the rounding function $\square$. More precisely, $\overline{N}$ can be defined as the set of rounding invariant properties of $\overline{M}$, i.e., it is $\overline{N} \subseteq \overline{M}$. Or in other words the structure $\{N,\overline{N}\}$ becomes a generalization of $\{M,\overline{M}\}$. If we move from the second to the third column in Fig. 1 we get a full generalization $\overline{N} \subset \overline{M}$. In the next and possibly further steps, $\overline{N} = \overline{M}$.

Let us now consider the question of how a given structure $\{M,\overline{M}\}$ can be approximated by a structure $\{N,\overline{N}\}$ with $N \subseteq M$. In a first approach one is tempted to try it with useful mapping properties like isomorphism and homomorphism. But it is easy to see that an isomorphism cannot be achieved and it can also be shown by simple examples in the case of the first line of Fig. 1 that a homomorphism cannot be realized in a sensible way. We shall see, however, that it is possible to implement in all cases a few necessary conditions for an homomorphism. With these conditions we come as close to a homomorphism as possible. Let us therefore first repeat the definition of a homomorphism.

Definition: Let $\{M,\overline{M}\}$ and $\{T,\overline{T}\}$ be two ordered algebraic structures and let a one-to-one correspondence exist between the operations and order relation(s) in M and T. Then a mapping $\square: M \to N$ is called a "homomorphism" if it is an algebraic homomorphism, i.e., if

$$\bigwedge_{a,b \in M} (\square a) \circledast (\square b) = \square(a * b) \tag{2}$$

for all corresponding operations $*$ and $\circledast$ and if it is an order homomorphism, i.e.,

$$\bigwedge_{a,b \in M} (a \leqq b \Rightarrow \square a \leqq \square b). \ \square \tag{3}$$

We are now going to derive these necessary conditions. If we restrict (2) to elements of N we immediately get, because of (R1),

$$\text{(R)} \quad \bigwedge_{a,b \in N} a \circledast b = \square(a * b).$$

Later we shall use this formula to define the operation $\circledast$, $* \in \{+,-,\cdot,/\}$, by the corresponding operation $*$ in M and the rounding $\square: M \to N$.

From (3) we immediately get that the rounding has to be a monotone function

$$\text{(R2)} \quad \bigwedge_{a,b \in M} (a \leqq b \Rightarrow \square a \leqq \square b), \qquad \text{monotone.}$$

If we further, in case of multiplication in (2), replace a by the negative multiple unit $-e$, we get

$$\bigwedge_{b \in M} \square(-b) = \underset{\text{(S),(R1)}}{\square(-e) \boxdot \square b} = (-e) \boxdot \square b = \underset{\text{(R)}\quad\text{(S),(R1)}}{= \square(-\square b)} = -\square b,$$

i.e.,

$$\text{(R3)} \quad \bigwedge_{a \in M} \square(-a) = -\square a, \qquad \text{antisymmetric.}$$

This means that the rounding has to be an antisymmetric function.

The conditions (R1),(R2),(R3) do not define the rounding function uniquely. We shall see later, however, that the structure of an ordered or weakly ordered ringoid or vectoid is invariant with respect to mappings with the properties (S), (R1), (R2), (R3), and (R). The proof of this assertion in all cases of Fig. 1 is a difficult task which cannot be solved within this paper. It is, however, essential that it can be given in all cases. (See [11]–[14], [16], [19], [20].)

Now there arises the question of whether an arithmetic which fulfills all our assumptions (R1),(R2),(R3),(R) can be implemented on computers in all cases of Fig. 1 by fast algorithms. We shall informatively answer this question positively within the next section. (For proofs, see [13], [14], [16], [3], [6].)

II. Further Roundings, Implementation, and Accuracy

The situation is the following. We have a set M with an operator $*$, for instance $+,-,\cdot,/$. On our computing tool in general the elements of M as well as the result of an operation $a * b$ are not exactly representable. Therefore we approximate the elements of M in a subset N by a proper rounding $\square: M \to N$. For an approximation of the operation $*$ we have derived the formula

$$\text{(R)} \quad \bigwedge_{a,b \in N} a \circledast b := \square(a * b).$$

At the first view this formula seems to contain a contradiction. The in general not representable result $a * b$ seems to be necessary for its realization. If, for instance, in the case of addition in a decimal floating-point system, a if of the magnitude 10^{50} and b of the magnitude 10^{-50}, about 100 decimal digits in the mantissa would be necessary for the representation of $a + b$. Even the largest computers do not have such long accumulators. A much more difficult situation arises in the case of a floating-point matrix multiplication or in the case of a division of complex floating-point numbers by formula (R). It can be shown, however, that in all cases in which $a * b$ is not representable on the computer it is sufficient to replace it by an appropriate and representable value $a \tilde{*} b$ with the property $\square(a * b) = \square(a \tilde{*} b)$. Then $a \tilde{*} b$ can be used to define $a \circledast b$ by

$$\bigwedge_{a,b \in T} a \circledast b := \square(a * b) = \square(a \tilde{*} b).$$

The proof of this assertion has to be given by concrete algorithms in all cases of Fig. 1.

Before we are going to discuss the question of implementation in more details let us increase the available set of roundings. A rounding $\square: M \to N$ is called "directed" if

$$\text{(R4)} \quad \bigwedge_{a \in M} \square a \leqq a, \qquad \text{downwardly directed}$$

$$\vee \bigwedge_{a \in M} a \leqq \square a, \qquad \text{upwardly directed.}$$

Let us now assume that the subset T of R in Fig. 1 is a floating-point system $T = T(\beta,n,e1,e2)$ wherein β denotes the base of the number system, n the number of digits in the mantissa, and $e1$ and $e2$ the least and greatest positive exponent. Then we shall use special notations for the following special roundings[1]:

∇a, monotone downwardly directed rounding

Δa, monotone upwardly directed rounding

$$\bigwedge_{a \geqq 0} \square_\beta a \leqq a \wedge \bigwedge_{a<0} \square_\beta a = -\square_\beta(-a),$$

monotone rounding towards zero

$$\bigwedge_{a \geqq 0} a \leqq \square_0 a \wedge \bigwedge_{a<0} \square_0 a = -\square_0(-a),$$

monotone rounding away from zero.

Further let

$$S_\mu(a) := \nabla a + \frac{\Delta a - \nabla a}{\beta} \cdot \mu, \qquad \mu = 1(1)\beta - 1. \quad (4)$$

Then we define roundings $\square_\mu: R \to T$, $\mu = 1(1)\beta - 1$, by

$$\bigwedge_{a \in [0,\beta^{e1-1})} \square_\mu a = 0$$

$$\bigwedge_{\beta^{e1-1} \leqq a \leqq B} \square_\mu^{1)} a = \begin{cases} \nabla a, & \text{for } a \in [\nabla a, S_\mu(a)) \\ \Delta a, & \text{for } a \in [S_\mu(a), \Delta a] \end{cases}$$

$$\bigwedge_{a<0} \square_\mu a = -\square_\mu(-a), \quad (5)$$

where $B := o \cdot (\beta - 1)(\beta - 1) \cdots (\beta - 1) \cdot \beta^{e2}$ denotes the greatest representable floating-point number.

If β is an even number, then $\square_{\beta/2}$ denotes the rounding to the nearest number of T and $\square_{\beta/2} a = (\nabla a - \Delta a)/2$.

The roundings $\{\nabla, \Delta, \square_\mu, \mu = o(1)\beta\}$ are not independent of each other. The following relations are easily verified:

$$\Delta a = -\nabla(-a) \quad (6)$$

$$\nabla a = -\Delta(-a) \quad (7)$$

$$\square_0 a = \text{sign } (a) \cdot \Delta|a|$$

$$\square_\beta a = \text{sign } (a) \cdot \nabla|a|.$$

All roundings $\square_\mu: R \to T$, $\mu = o(1)\beta$, are further antisymmetric functions. From (4)–(7) it follows that all these roundings can be expressed by the monotone downwardly (respectively, upwardly) directed rounding.

An algorithm for the realization of formula (R) can in principle be separated into the following five steps.

1) Decomposition of a and b, i.e., separation of a and b into exponent part and mantissa (DC).

2) Execution of the operation $a \tilde{*} b$. It is possible that $a \tilde{*} b = a * b$.

3) Normalization of $a \tilde{*} b$. If the result is already normalized this step can be omitted (N).

4) Rounding of $a \tilde{*} b$ to $a \circledast b = \square(a * b) = \square(a \tilde{*} b)$ (R).

5) Composition, i.e., combination of the resulting exponent part and mantissa to a floating-point number (C).

Fig. 2 gives a graphical diagram of these five steps. A more detailed discussion of these steps can be found in the literature [13]–[15], [17], [22], [8].

The algorithms can be implemented using accumulators of different lengths. A convenient algorithm uses an accumulator of one digit which can be a binary digit in front and $2n + 1$ digits of base β after the point. See Fig. 3. A more structured algorithm does it with an accumulator with one digit which can be a binary digit in front of the point and $n + 2$ digits of base β plus one binary digit after the point. See Fig. 3. This algorithm shows that a further reduction of the length of the accumulator is impossible if formula (R) is to be strictly realized.

The algorithms show as an essential result that the whole implementation can be separated into five steps, as indicated above, which are independent of each other. This means that the provisional result $a \tilde{*} b$ can be chosen independently of the rounding function such that for all $* \in \{+,-,\cdot,/\}$ and for all roundings of the set $\{\nabla, \Delta, \square_\mu, \mu = 0(1)\beta\}$, formula (R) holds.

With these algorithms the question of implementation

[1] Since it is not necessary for the purpose of this paper we do not define the roundings $\square_\mu$, $\mu = 1(1)\beta$, for $|a| > B$.

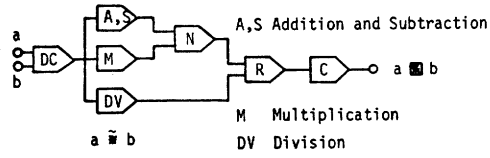


Fig. 2. Flow diagram for the arithmetic operations.

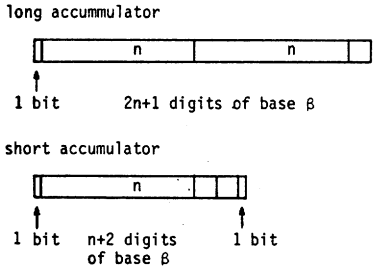


Fig. 3. Long and short accumulators.

is solved for the case of the first line of Fig. 1 and, since formula (R) has also been implemented with the roundings ∇ and Δ, also for most of the interval lines in Fig. 1. This last assertion will be discussed later.

We are now going to discuss briefly the implementation in case of matrix operations. Let $\square: R \to T$ be a rounding. If we define a mapping $\square: MR \to MT$ by

$$\bigwedge_{A=(a_{ij})\in MR} \square A := (\square a_{ij})$$

then $\square: MR \to MT$ is also a rounding. If, further, the rounding $\square: R \to T$ is monotone, directed, antisymmetric, respectively, then the rounding $\square: MR \to MT$ is also monotone, directed, antisymmetric, respectively.

By formula (R) the operations $\boxed{*}$, $* \in \{+, \cdot\}$, in MT have to be defined by

$$\text{(R)} \qquad \bigwedge_{A,B\in MT} A \boxed{*} B := \square(A * B),$$

for all $* \in \{+, \cdot\}$.

If $A = (a_{ij})$ and $B = (b_{ij})$, then we get in case of addition

$$A \boxplus B := \square(A + B) = (a_{ij} \boxplus b_{ij}).$$

Here, the addition on the right-hand side means the addition in T which by assumption is properly defined and there is no problem connected with the addition.

In the case of multiplication, however, we get

$$A \boxdot B := \square(A \cdot B) := \square\left(\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}\right) \tag{8}$$

where in

$$\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j} \tag{9}$$

the multiplications and additions denote the real multiplication and addition. Even on computers with a so-called accumulator of double length only in very rare cases is (9) exactly representable. The algorithms show, however, that whenever (9) is not representable it can be replaced by an appropriate and representable value

$$\widetilde{\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}} \tag{10}$$

with the property

$$\square(A \cdot B) = \square\left(\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}\right) = \square(A \tilde{\cdot} B) = \square\left(\widetilde{\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}}\right). \tag{11}$$

Then (11) can be used to define (8). The explicit algorithms prove this assertion. See [16], [3]. In order to realize (11) at first the products $a_{i\nu} \cdot b_{\nu j}$ are calculated. If a_{ij} and b_{ij} are floating-point numbers of n digits in the mantissa, then $a_{i\nu} \cdot b_{\nu j}$ can exactly be generated within an accumulator of $L = 2n$ digits. If this is done then (11) can be generated if the sum

$$z := \square\left(\sum_{i=1}^{r} x_i\right) = \square\left(\widetilde{\sum_{i=1}^{r} x_i}\right) \tag{12}$$

can be implemented where the x_i, $i = 1(1)r$, denote $L = 2n$ digit floating-point numbers and z is an n digit floating-point number. The algorithms mentioned above could also be used to produce a floating-point number z defined by (12) of $n, n+1, \cdots, L = 2n$ correct digits just by rounding the intermediate result $\widetilde{\sum_{i=1}^{r} x_i}$ to another length. These algorithms again can be separated into several independent steps which means that the intermediate result $\sum_{i=1}^{r} x_i$ can be chosen independently of the rounding function such that for all roundings of the set $\square \in \{\nabla, \Delta, \square_\mu, \mu = o(1)\beta\}$ the equality

$$\square\left(\sum_{i=1}^{r} x_i\right) = \square\left(\widetilde{\sum_{i=1}^{r} x_i}\right)$$

holds. The whole algorithm uses an accumulator with one digit which can be a binary digit in front of the point and $L + 2$ digits of base β plus one further binary digit after the point. If n denotes the number of digits of the floating-point mantissa then $L = 2n$. See Fig. 4.

With this algorithm the question of implementation is solved not only in case of the third line of Fig. 1 but also in the cases of vector matrix multiplication, multiplication of complex floating-point numbers by formula (R), complex floating-point matrix products, and matrix vector multiplication, and, since formula (R) has also been realized for the roundings ∇ and Δ, in all cases of interval structures occurring in Fig. 1.

As far as the implementation is concerned there remains only one open question. This is the case of complex floating-point division. In this case the formula

$$\square\left(\frac{ab + cd}{ef + gh}\right)$$

has to be realized. But this problem has also been solved in [6]. In this case still a slightly longer accumulator is necessary. The running time for a software solution of this

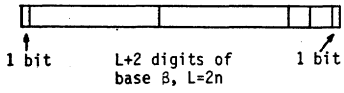


Fig. 4. Length of the accumulator for scalar products.

quotient compared with the usual complex quotient (UNIVAC 1108) was enlarged by an average factor of 1.2. If we take into account the improvements with respect to error analysis (see below) or to a much better theoretical understanding of computer arithmetic (see below) this shows that such algorithms should be realized.

Let us, for the case of matrix operations, still discuss the general advantage which we get if we define the computer arithmetic by formula (R) in all lines of Fig. 1. Fig. 5(a) describes the way in which matrix operations on computers are usually defined. The matrix operations in MT for instance are defined by the floating-point operations in T and the usual formulas for matrix addition and multiplication of real matrices. An error analysis of such an arithmetic has to go back to the elementary floating-point operation and in general there are no obvious compatibility properties valid between the matrix operations in MR and MT.

Fig. 5(b) describes the new way of defining floating-point matrix operations by formula (R). The operations in MT, for instance, are directly defined by the operations in MR. This leads to a much higher accuracy and allows a much simpler error analysis (see below). Further, by the rounding properties (R1),(R2),(R3) which we have assumed, the following reasonable compatibility properties between the structure in MT and that in MR are easily verified:

$$\text{(RG1)} \quad \bigwedge_{A,B\in MT} (A * B \in MT) \Rightarrow A \boxed{*} B = A * B), \qquad \text{for all } * \in \{+,-,\cdot\}$$

$$\text{(RG2)} \quad \bigwedge_{A,B,C,D\in MT} (A * B \leqq C * D \Rightarrow A \boxed{*} B \leqq C \boxed{*} D), \qquad \text{for all } * \in \{+,-,\cdot\}$$

$$\text{(RG3)} \quad \bigwedge_{A\in MT} -A = \boxminus A := (-E) \boxdot A, \qquad E \text{ unit matrix.}$$

(RG1) should be valid for every computer arithmetic, (RG2) expresses its monotonicity, and (RG3) the identity of the minus operators in MR and MT.

In all interval lines in Fig. 1 the rounding is furthermore upwardly directed. Then we get a fourth compatibility property:

$$\text{(RG4)} \quad \bigwedge_{A,B} A * B \leqq A \boxed{*} B.$$

In this case the $\leqq$ sign means the inclusion and (RG4) then says that the result of an operation in the original set is always included in the result in the approximating subset.

Concerning accuracy we begin with the following well known result: Let $T = T(\beta,n,e1,e2)$ be a floating-point system and $\square: R \to T$ a monotone rounding and let $\delta(\square a) := a - \square a$, denote the absolute rounding error and $\epsilon := \delta(\square a)/a$ the relative rounding error. Then

$$\bigwedge_{a\in R} (b^{e1-1} \leqq |a| < b^{e2} \Rightarrow \square a = a(1-\epsilon) \text{ with } |\epsilon| < \epsilon^* \Rightarrow |a - \square a| \leqq \epsilon^* \cdot |a|)$$

where

$$\epsilon^* := \begin{cases} \frac{1}{2}\beta^{1-n}, & \text{for the rounding to the nearest floating-point number} \\ \beta^{1-n}, & \text{else.} \end{cases} \tag{13}$$

If we define floating-point arithmetic by formula (R) and a monotone and antisymmetric rounding we immediately get, for all operations $* \in \{+,-,\cdot,/\}$,

$$\bigwedge_{a,b\in T} (\beta^{e1-1} \leqq |a * b| < \beta^{e2} \Rightarrow a \boxed{*} b = (a * b)(1-\epsilon), \text{ with } |\epsilon| < \epsilon^* \Rightarrow |a * b - a \boxed{*} b| \leqq \epsilon^* \cdot |a * b|)$$

where ϵ^* is defined by (13).

This result is the base for most rounding error estimations in numerical mathematics. It should, however, be clear that such estimations only lead to reliable error *bounds* if formula (R) is strictly implemented.

Error estimations for floating-point matrix computations are usually derived in the sense of Fig. 5(a). See [21]. If we apply the new definition (R) [Fig. 5(b)] we get identically the same formulas as in the case of the elementary floating-point operations. Again let $\square: R \to T$ be a monotone and antisymmetric rounding and let a rounding $\square: MR \to MT$ be defined by

$$\bigwedge_{A=(a_{ij})\in MR} \square A := (\square a_{ij}).$$

Then

$$\bigwedge_{A=(a_{ij})\in MR} \left(\bigwedge_{i,j} b^{e1-1} \leqq |a_{ij}| < b^{e2} \Rightarrow \square A = (a_{ij}(1-\epsilon_{ij})), \text{ with } |\epsilon_{ij}| < \epsilon^* \Rightarrow |A - \square A| \leqq \epsilon^* \cdot |A| \right)$$

where ϵ^* is defined by (13) and the absolute value is defined componentwise.

If in MT operations $\boxed{*}$ $* \in \{+,\cdot\}$, are defined by formula (R) and $A,B \in MT$ we get with the abbreviation $Z := (Z_{ij}) := A * B$ for all operations $* \in \{+,-,\cdot\}$:

$$\bigwedge_{A,B\in MT} \left(\bigwedge_{i,j} b^{e1-1} \leqq z_{ij} < b^{e2} \Rightarrow A \boxed{*} B = (z_{ij}(1-\epsilon_{ij})) \quad \text{with } |\epsilon_{ij}| < \epsilon^* \Rightarrow |A * B - A \boxed{*} B| \leqq \epsilon^* \cdot |A * B| \right). \tag{14}$$

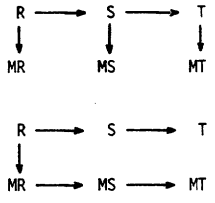


Fig. 5. Definition of floating-point matrix operations.

This is the same simple formula with the same ϵ^* which we have gotten in case of the elementary floating-point operations. Because of its much simpler form it allows a much simpler error analysis for floating-point matrix computations than an error analysis derived in the sense of Fig. 5(a). Furthermore (14) is more accurate. In [5] an error analysis of the Gauss algorithm for linear equations using formula (14) is given. See also [4].

Contrary to most error estimations in numerical mathematics, the error formulas derived in this paper lead to absolute error *bounds* if formula (R) is strictly implemented.

III. The Structure of Computer Arithmetic

In the literature several attempts to formalize computer arithmetic are known. All these models are only interested in describing the relationship between the real numbers and a floating-point system. It turns out, however, that the real numbers have too many very special properties for us to recognize all essential properties already at this stage. Only the entirety of the structures listed in Fig. 1 seems to give the frame which allows a general theory of computations in subsystems. Essential contributions towards a theoretical understanding come especially from interval arithmetic. Roughly it can be said that between the power set of an ordered algebraic structure and its intervals there exists mathematically the same relationship as between the real numbers and a floating-point system.

An abstract theory of computations in subsets has to begin with a characterization of the essential properties of the sets in Fig. 1. All these sets are ordered with respect to certain order relations. Let us consider the interval vectors of dimension 2, IV_2R. These are intervals of two-dimensional real vectors. Geometrically such a vector describes a rectangle with sides parallel to the axes. These interval vectors are special elements of the power set PV_2R of the real vectors which is defined as the set of all subsets of real vectors. Between these sets the following relationship holds:

1) For all $a \in PV_2R$ there exist upper bounds (with respect to the inclusion as order relation) in the subset IV_2R. See Fig. 6.

2) For all $a \in PV_2R$ the set of all upper bounds in the subset IV_2R has a least element. See Fig. 6.

These two properties also characterize the relationship between any set of Fig. 1 and its subset(s) on the right-hand side. Let us now consider the set of real numbers R

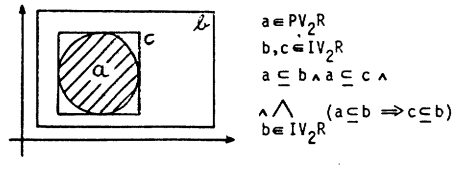


Fig. 6. To the concept of a screen.

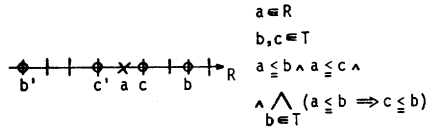


Fig. 7. To the concept of a screen.

and a subset t of floating-point numbers. We have again the two properties.

1) For all $A \in R$ there exist upper bounds (with respect to the order relation $\leqq$ of the real numbers) in the subset T. See Fig. 7.

2) For all $a \in R$ the set of all upper bounds in the subset T has a least element. See Fig. 7.

In this case corresponding properties are also valid for the lower bounds. We summarize these properties by the following definition.

Definition: Let $\{M, \leqq\}$ be an ordered set and $L(a) := \{b \in M | b \leqq a\}$, respectively $U(a) := \{b \in M | a \leqq b\}$, denote the set of all lower, respectively upper bounds of a. A subset $T \subseteq M$ is called a lower, respectively an upper, screen of M if

$$\text{(S1)} \quad \bigwedge_{a \in M} L(a) \cap T \neq \phi,$$

$$\text{respectively} \bigwedge_{a \in M} U(a) \cap T \neq \phi$$

$$\text{(S2)} \quad \bigwedge_{a \in M} \bigvee_{x \in L(a) \cap T} \bigwedge_{b \in L(a) \cap T} b \leqq x,$$

$$\text{respectively} \bigwedge_{a \in M} \bigvee_{x \in U(a) \cap T} \bigwedge_{b \in U(a) \cap T} x \leqq b.$$

If $T \subseteq M$ is simultaneously a lower and an upper screen, then it is called a screen of M [9]. □

In all essential applications of the concept of a screen the basic set M is not only an ordered set but a complete lattice. In this case necessary and sufficient conditions for the concept of a screen can be derived. See [10].

Let $\{M, \leqq\}$ be a complete lattice with the greatest element $i(M)$ and the least element $o(M)$. If a subset $T \subseteq M$ is also a complete lattice it is called a complete subformation. Then

$$\bigwedge_{A \subseteq T, A \neq \phi} (\inf_T A \leqq \inf_M A \wedge \sup_M A \leqq \sup_T A).$$

If the first inequality always is an equality T is called a complete infimum-subformation of M and in the dual case a complete supremum-subformation of M. If both ine-

qualities are always equalities T is called a complete sublattice of M. The following theorem holds.

Theorem: A subset $\{T,\leqq\}$ of a complete lattice $\{M,\leqq\}$ is a lower screen (respectively an upper screen) of $\{M,\leqq\}$ if and only if

(S1′) $o(M) = o(T)$ (respectively $i(M) = i(T)$)

and

(S2′) $\{T,\leqq\}$ is a complete supremum-subformation (respectively) a complete infimum-subformation) of $\{M,\leqq\}$.

$\{T,\leqq\}$ is a screen of $\{M,\leqq\}$ if and only if $o(M) = o(T)$, $i(M) = i(T)$, and $\{T,\leqq\}$ is a complete sublattice of $\{M,\leqq\}$. □

For the proof see [10]. Now it can be shown that all sets in Fig. 1 are screens of the set(s) on their left-hand side. See [11]–[13].

With this concept further theorems can be derived. For instance if $\{M,\leqq\}$ is a complete lattice and $\{T,\leqq\}$ a lower, respectively an upper, screen then the monotone downwardly respectively upwardly, directed rounding can be characterized by

$$\bigwedge_{a\in M} \nabla a = \sup\,(L(a) \cap T),$$

$$\text{respectively } \bigwedge_{a\in M} \Delta a = \inf\,(U(a) \cap T).$$

If further $\{M,*\}$ is a groupoid with a right neutral element then $\{T,\circledast\}$ is a groupoid on the screen with the properties (RG1),(RG2),(RG4) if and only if

$$\bigwedge_{a,b\in T} a \circledast b = \nabla(a * b),$$

$$\text{respectively } \bigwedge_{a,b\in T} a \circledast b = \Delta(a * b).$$

For proofs and applications see [9], [13].

We are now going to define the special structures of a weakly ordered, respectively an ordered, ringoid and derive its most important properties. We shall later see that this, under the assumptions (S1), (S2), (S), (R1), (R2), (R3), and (R), describes the structures in the lines 1,3,4,6,7,9,10,12 of Fig. 1.

Definition: A nonempty set R in which an addition and a multiplication are defined is called a ringoid if

(D1) $\bigwedge_{a,b\in R} a + b = b + a$

(D2) $\bigvee_{o\in R} \bigwedge_{a\in R} a + o = a$

(D3) $\bigvee_{e\in R\setminus\{o\}} \bigwedge_{a\in R} a \cdot e = e \cdot a = a$

(D4) $\bigwedge_{a\in R} a \cdot o = o \cdot a = o.$

(D5) There exists an element $x \in R\setminus\{e\}$ such that
(a) $x \cdot x = e$

(b) $\bigwedge_{a,b\in R} x(ab) = (xa)b = a(xb)$

(c) $\bigwedge_{a,b\in R} x(a + b) = xa + xb$

(D6) x is unique.

If furthermore in a ringoid a division/: $R \times R\setminus N \to R$ is defined with $N \subseteq R$ and $o \in N$ it is called a division-ringoid if

(D7) $\bigwedge_{a\in R} a/e = a$

(D8) $\bigwedge_{a\in R\setminus N} o/a = o$

(D9) Besides (D5) the element x also fulfils the property

$$\bigwedge_{a\in R} \bigwedge_{b\in R\setminus N} x(a/b) = (xa)/b = a/(xb).$$

A ringoid is called weakly ordered if $\{R,\leqq\}$ is an ordered[2] set and

(OD1) $\bigwedge_{a,b,c\in R} (a \leqq b \Rightarrow a + c \leqq b + c)$

(OD2) $\bigwedge_{a,b\in R} (a \leqq b \Rightarrow -b \leqq -a).$

A weakly ordered ringoid, respectively division-ringoid, is called an ordered ringoid, respectively an ordered division-ringoid, if

(OD3) $\bigwedge_{a,b,c\in R} (o \leqq a \leqq b \wedge c \geqq o \Rightarrow a \cdot c \leqq b \cdot c \wedge c \cdot a \leqq c \cdot b)$, respectively

(OD4) $\bigwedge_{a,b,c\in R} (o < a \leqq b \wedge c > o \Rightarrow o \leqq a/c \leqq b/c \wedge c/a \geqq c/b \geqq o).$ □

The uniqueness of x can be used for the following definition.

Definition: In a ringoid R we define a minus operator and a subtraction by

$$\bigwedge_{a\in R} -a := x \cdot a \tag{15}$$

$$\bigwedge_{a,b\in R} a - b := a + (-b). \quad \square \tag{16}$$

[2] $\{R,\leqq\}$ is an ordered set means $\leqq$ is a reflexive (O1), transitive (O2), and antisymmetric (O3) relation.

Simple consequences:

(1) $\underset{a=e}{\Rightarrow} x = -e$

(D5a) $\Rightarrow (-e)(-e) = e$

(D5b) $\Rightarrow -(ab) = (-a)b = a(-b)$

(D5c) $\Rightarrow -(a + b) = (-a) + (-b)$

(OD2) $\Rightarrow (a \leqq b \Rightarrow -b \leqq -a)$.

In general additive inverses do not exist within a ringoid. But nevertheless the subtraction is no independent operation. It is defined by the multiplication and the addition.

Theorem: In a ringoid R the following properties hold:

(a) $e \neq o$, $-e \neq o$, $-e \neq e$.

(b) $o - a = -a$

(c) $-a = (-e) \cdot a = a \cdot (-e)$

(d) $-(-a) = a$

(e) $-(a - b) = -a + b = b - a$

(f) $(-a)(-b) = ab$

(g) o, respectively e, is the only neutral element of the addition, respectively multiplication

(h) o is the only right neutral element of the subtraction.

In a division-ringoid we get further

(i) $(-a)/(-b) = a/b$

(j) $(-e)/(-e) = e$.

In a weakly ordered ringoid holds

(k) $a \leqq b \wedge c \leqq d \Rightarrow a + c \leqq b + c$

(l) $a < b \Rightarrow -b < -a$.

In an ordered ringoid, respectively ordered division ringoid, we get

(m) $o \leqq a \leqq b \wedge o \leqq c \leqq d \Rightarrow o \leqq ac \leqq bd \wedge o \leqq ca \leqq db$

(n) $a \leqq b \leqq o \wedge c \leqq d \leqq o \Rightarrow o \leqq bd \leqq ac \wedge o \leqq db \leqq ca$

(o) $a \leqq b \leqq o \wedge o \leqq c \leqq d \Rightarrow ad \leqq bc \leqq o \wedge da \leqq cb \leqq o$

(p) $a > o \wedge b > o \Rightarrow a/b \geqq o$

(q) $a < o \wedge b > o \Rightarrow a/b \leqq o \wedge b/a \leqq o$

(r) $a < o \wedge b < o \Rightarrow a/b \geqq o$. □

The proof is left to the reader. See [11], [13]. The theorem can be summarized. In a ringoid the same rules for the minus operator hold as in the real number field. In an ordered ringoid for all elements which are comparable with o with respect to $\leqq$ and $\geqq$ the same rules for inequalities hold as in the real number field.

Applications: Let R be a division-ringoid.

If MR denotes the set of $r \times r$ matrices with components out of R and in MR the equality, addition, and multiplication are defined by the usual formulas for the components then MR is also a ringoid.

If PR denotes the power set of R and in PR operations are defined by (1) then PR also becomes a division-ringoid.

Let CR denote the set of pairs of elements of R and let in CR an addition, multiplication, and division be defined by the same formulas as in the complex number field; then CR also becomes a division-ringoid.

If R is a weakly ordered division-ringoid and in MR, respectively CR, an order relation is defined componentwise then MR is a weakly ordered ringoid, respectively CR a weakly ordered division-ringoid.

If furthermore R is an ordered ringoid than MR is also an ordered ringoid.

For the proofs of these results see [11] and [23].

If in Fig. 1 R is an ordered division-ringoid then by these results the structure is also known in the first elements of the lines 3, 4, 6, 7, 9, 10, and 12.

We are now going to discuss the theorems which allow us to transfer these structures to the subsets on the right-hand side.

Theorem: Let R be a ringoid with the special elements $\{-e,o,e\}$, $\{R,\leqq\}$ a complete lattice, and $\{T,\leqq\}$ a symmetric screen (S1),(S2),(S) (respectively, a symmetric lower screen respectively a symmetric upper screen), $\square: R \to T$ an antisymmetric rounding (R1),(R3), and let in T operations $\boxed{*}$, $* \in \{+, \cdot\}$, be defined by formula (R).

1) In T the following properties hold: (D1), (D2) for o, (D3) for e, (D4), (D5) for $-e$, and

$$\text{(RG1)} \quad \bigwedge_{a,b \in T} (a * b \in T \Rightarrow a \boxed{*} b = a * b), \qquad * \in \{+,-,\cdot\}$$

$$\text{(RG3)} \quad \bigwedge_{a \in T} -a = (-e) \boxdot a.$$

2) If $\square: R \to T$ is monotone (R2) $\Rightarrow$

$$\text{(RG2)} \quad \bigwedge_{a,b,c,d \in T} (a * b \leqq c * d \Rightarrow a \boxed{*} b \leqq c \boxed{*} d), \qquad * \in \{+,-,\cdot\}.$$

3) If $\square: R \to T$ is upwardly, respectively downwardly, directed (R4) $\Rightarrow$

$$\text{(RG4)} \quad \bigwedge_{a,b \in T} a \boxed{*} b \leqq a * b, \text{ respectively}$$

$$\bigwedge_{a,b \in T} a * b \leqq a \boxed{*} b, \qquad * \in \{+,-,\cdot\}.$$

4) If R is weakly ordered (OD1), (OD2) and $\square: R \to T$ monotone $\Rightarrow$ T is weakly ordered, i.e., (OD1), (OD2) hold.

5) If R is ordered (OD3) and $\square: R \to T$ monotone $\Rightarrow$ in T (OD3) holds. □

Theorem: Let R be a division-ringoid with the special elements $\{-e,o,e\}$, $\{R,\leqq\}$ a complete lattice, and $\{T,\leqq\}$ a symmetric screen (respectively a symmetric lower screen, respectively a symmetric upper screen), $\square: R \to T$ an antisymmetric rounding and let in T operations $\boxed{*}$, $* \in \{+, \cdot, /\}$ be defined by formula (R).

1) In T the following properties hold: (D1), (D2) for o,

(D3) for e, (D4), (D5) for $-e$, (D7), (D8), (D9) for $-e$, (RG1) for $* \in \{+,-,\cdot,/\}$ and (RG2).

2) If $\square: R \to T$ is monotone $\Rightarrow$ (RG2) for $* \in \{+,-,\cdot,/\}$.

3) If $\square: R \to T$ is downwardly respectively upwardly, directed $\Rightarrow$ (RG4) for $* \in \{+,-,\cdot,/\}$.

4) If R is an ordered division-ringoid and $\square: R \to T$ monotone $\Rightarrow$ in T (OD4) holds. □

All statements of these theorems are easily verified. As an example we prove the properties (D5c) and (OD1):

$$\text{(D5c):}\quad (-e) \boxdot a = \square(-a) = -\square a = -a \in T \qquad (17)$$
$$\text{(R)}\quad \text{(R3)}\quad \text{(R1)}\quad \text{(S)}$$
$$(-e) \boxdot (a \boxplus b) = \square((-e)\cdot\square(a+b)) =$$
$$\text{(R)}\qquad\qquad \text{(R3)}$$
$$= \square(\square(-(a+b))) = \square(-(a+b)) =$$
$$\text{(R1)}\qquad\qquad (\text{D5c})_R$$
$$= \square((-a)+(-b)) = (-a)\boxplus(-b) =$$
$$\text{(R)}\qquad\qquad \text{(3)}$$
$$= ((-e)\boxdot a)\boxplus((-e)\boxdot b).$$

$$\text{(OD1):}\quad a \leqq b \Rightarrow a + c \leqq b + c \Rightarrow \square(a+c)$$
$$(\text{OD1})_R \qquad \text{(R2)}$$
$$\leqq \square(b+c) \Rightarrow$$
$$\text{(R)}$$
$$\Rightarrow a \boxplus c \leqq b \boxplus c.$$

The proofs of these two properties show already that our assumptions (S),(R1),(R2),(R3),(R) are really necessary in order to get the desired structure in T. If we change these properties or do not realize them strictly we get a different structure in the subset T.

The last two theorems show that if we proceed as stated we get nearly again the structure of a ringoid in the subset T. The only property which cannot be proved by a general theorem is (D6). The proof of this property is a difficult task in all cases of Fig. 1. Concerning to these proofs we refer to the literature [11]–[14], [16], [20].

We still indicate the proof in the case of the first line of Fig. 1. As usual we call an ordered set linearly ordered if (O4) holds:

$$\text{(O4)}\quad \bigwedge_{a,b\in R} (a \leqq b \vee b \leqq a).$$

Theorem: In case of a linearly ordered set $\{R,\leqq\}$ (D6) is no independent assumption, i.e., (O1),(O2),(O3),(O4),(D1),(D2),(D3),(D4),(D5),(OD1),(OD2),(OD3) $\Rightarrow$ (D6). □

This theorem guarantees that the structure of the floating-point numbers S and T (first line of Fig. 1) is that of a linearly ordered division-ringoid.

We are now going to define the structure of the "higher dimensional spaces" listed in Fig. 1. We shall later see that the structure of a weakly ordered, respectively an ordered, vectoid under the assumptions (S1),(S2),(S),(R2),(R3) and (R) describes the structures in the lines 2,3,5,6,8,9,11,12 of Fig. 1.

Definition: Let R be a ringoid with elements a,b,c, $\cdots$ and the special elements $\{-e,o,e\}$ and $\{V,+\}$ a groupoid with elements $\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\cdots$ and the properties

$$\text{(V1)}\quad \bigwedge_{\boldsymbol{a},\boldsymbol{b}\in V} \boldsymbol{a}+\boldsymbol{b} = \boldsymbol{b}+\boldsymbol{a}$$

$$\text{(V2)}\quad \bigvee_{\boldsymbol{o}\in V}\bigwedge_{\boldsymbol{a}\in V} \boldsymbol{a}+\boldsymbol{o} = \boldsymbol{a}.$$

V is called an R-vectoid $\{V,R\}$ if there is a multiplication $\cdot: R \times V \to V$ which, when defined, with the abbreviation

$$\bigwedge_{\boldsymbol{a}\in V} -\boldsymbol{a} := (-e)\cdot\boldsymbol{a},$$

fulfills the following properties:

$$\text{(VD1)}\quad \bigwedge_{a\in R}\bigwedge_{\boldsymbol{a}\in V} (a\cdot\boldsymbol{o} = \boldsymbol{o} \wedge o\cdot\boldsymbol{a} = \boldsymbol{o})$$

$$\text{(VD2)}\quad \bigwedge_{\boldsymbol{a}\in V} e\cdot\boldsymbol{a} = \boldsymbol{a}$$

$$\text{(VD3)}\quad \bigwedge_{a\in R}\bigwedge_{\boldsymbol{a}\in V} -(a\cdot\boldsymbol{a}) = (-a)\cdot\boldsymbol{a} = a\cdot(-\boldsymbol{a})$$

$$\text{(VD4)}\quad \bigwedge_{\boldsymbol{a},\boldsymbol{b}\in V} -(\boldsymbol{a}+\boldsymbol{b}) = (-\boldsymbol{a})+(-\boldsymbol{b}).$$

An R-vectoid is called "multiplicative" if in V also a multiplication $\cdot: V \times V \to V$ is defined with the properties:

$$\text{(V3)}\quad \bigvee_{\boldsymbol{e}\in V\setminus\{\boldsymbol{o}\}}\bigwedge_{\boldsymbol{a}\in V} \boldsymbol{a}\cdot\boldsymbol{e} = \boldsymbol{e}\cdot\boldsymbol{a} = \boldsymbol{a}$$

$$\text{(V4)}\quad \bigwedge_{\boldsymbol{a}\in V} \boldsymbol{a}\cdot\boldsymbol{o} = \boldsymbol{o}\cdot\boldsymbol{a} = \boldsymbol{o}$$

$$\text{(VD5)}\quad \bigwedge_{\boldsymbol{a},\boldsymbol{b}\in V} -(\boldsymbol{a}\boldsymbol{b}) = (-\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a}(-\boldsymbol{b}).$$

An R-vectoid is called "weakly ordered" $\{V,R,\leqq\}$ if $\{V,\leqq\}$ is an ordered set and

$$\text{(OV1)}\quad \bigwedge_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\in V} (\boldsymbol{a}\leqq\boldsymbol{b}\Rightarrow\boldsymbol{a}+\boldsymbol{c}\leqq\boldsymbol{b}+\boldsymbol{c})$$

$$\text{(OV2)}\quad \bigwedge_{\boldsymbol{a},\boldsymbol{b}\in V} (\boldsymbol{a}\leqq\boldsymbol{b}\Rightarrow -\boldsymbol{b}\leqq -\boldsymbol{a})$$

A weakly ordered vectoid is called "ordered" if R is an ordered ringoid and

$$\text{(OV3)}\quad \bigwedge_{a,b\in R}\bigwedge_{\boldsymbol{a},\boldsymbol{b}\in V} (o\leqq a\leqq b \wedge \boldsymbol{o}\leqq\boldsymbol{a}\Rightarrow a\cdot\boldsymbol{a}$$
$$\leqq b\cdot\boldsymbol{a} \wedge$$
$$o\leqq a \wedge \boldsymbol{o}\leqq\boldsymbol{a}\leqq\boldsymbol{b}\Rightarrow a\cdot\boldsymbol{a}\leqq a\cdot\boldsymbol{b}).$$

A multiplicative vectoid is called "weakly ordered" if it is a weakly ordered vectoid. A multiplicative vectoid is called "ordered" if it is an ordered vectoid and

$$\text{(OV4)}\quad \bigwedge_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\in V} (\boldsymbol{o}\leqq\boldsymbol{a}\leqq\boldsymbol{b} \wedge \boldsymbol{o}\leqq\boldsymbol{c}\Rightarrow\boldsymbol{a}\cdot\boldsymbol{c}\leqq\boldsymbol{b}\cdot\boldsymbol{c}$$
$$\wedge\ \boldsymbol{c}\cdot\boldsymbol{a}\leqq\boldsymbol{c}\cdot\boldsymbol{b}).$$ □

Definition: In a vectoid we define a subtraction by

$$\bigwedge_{a,b\in V} a - b := a + (-b). \qquad \square$$

Again in general there do not exist inverse elements of the addition within a vectoid. But nevertheless the subtraction is no independent operation. It is defined by the multiplication with elements of R and the addition.

Theorem: In a vectoid $\{V,R\}$ the following properties hold.

(a) o is the only neutral element of the addition.
(b) $o - a = -a$.
(c) $-(-a) = a$.
(d) $-(a - b) = -a + b = b - a$.
(e) $(-a)(-a) = a \cdot a$.
(f) $-a = o \Leftrightarrow a = o$.

In a multiplicative vectoid $\{V,R\}$ we get further

(g) e is the only neutral element of the multiplication.
(h) $-a = (-e) \cdot a = a \cdot (-e)$.
(i) $(-a) \cdot (-b) = a \cdot b$.

In a weakly ordered vectoid the following hold.

(j) $a \leqq b \wedge c \leqq d \Rightarrow a + c \leqq b + d$
(k) $a < b \Rightarrow -b < -a$.

In an ordered vectoid, respectively ordered multiplicative vectoid, we get the following.

(l) $o \leqq a \leqq b \wedge o \leqq c \leqq d \Rightarrow o \leqq ac \leqq bd$
(m) $a \leqq b \leqq o \wedge c \leqq d \leqq o \Rightarrow o \leqq bd \leqq ac$
(n) $a \leqq b < o \wedge o \leqq c \leqq d \Rightarrow ad \leqq bc \leqq o$
(o) $o \leqq a \leqq b \wedge c \leqq d \leqq o \Rightarrow bc \leqq ad \leqq o$
(p) $o \leqq a \leqq b \wedge o \leqq c \leqq d \Rightarrow o \leqq ac \leqq bd \wedge o \leqq ca \leqq db$
(q) $a \leqq b \leqq o \wedge o \leqq c \leqq d \Rightarrow ad \leqq bc \leqq o \wedge da \leqq cb \leqq o$
(r) $a \leqq b \leqq o \wedge c \leqq d \leqq o \Rightarrow o \leqq bd \leqq ac \wedge o \leqq db \leqq ca$. $\square$

The proof is left to the reader. See [19], [13]. The theorem can be summarized. In a vectoid the same rules for the minus operator hold as in the real vector space. In an ordered vectoid for all elements which are comparable with o with respect to $\leqq$ and $\geqq$ the same rules for inequalities hold as in the real vector space.

Applications

Let $\{V,R\}$ be a vectoid. Then the power set $\{PV,PR\}$ is a vectoid as well as $\{PV,R\}$ is a vectoid.

Let R be a ringoid with the special elements $\{-e,o,e\}$.

If $VR := R \times R \times \cdots \times R$ denotes the set of vectors with components out of R and in VR the equality, addition and multiplication by elements of R are defined by the usual formulas for the components then $\{VR,R\}$ is a vectoid.

If MR denotes the set of $r \times r$ matrices with components out of R and in MR the equality, addition, and multiplication as well as the multiplication by elements of R are defined by the usual formulas for the components then $\{MR,R\}$ is a multiplicative vectoid.

If VR again denotes the set of n-tuples over R and in VR the equality, addition, and multiplication by elements out of MR are defined by the usual formulas for the components then $\{VR,MR\}$ is a vectoid.

If R is a weakly ordered, respectively an ordered, ringoid then also $\{VR,R,\leqq\}$ as well as $\{VR,MR,\leqq\}$ are weakly ordered, respectively ordered, vectoids. $\{MR,R\leqq\}$ is a weakly ordered, respectively an ordered, multiplicative vectoid.

The proof of these results is left to the reader. See [19] and [13] or [23].

If in Fig. 1 R is an ordered ringoid then by these results the structure is also known in the first elements of the lines 2, 3, 5, 6, 8, 9, 11, and 12.

We are now going to discuss the theorems which allow us to transfer these structures to the subsets on the right-hand side.

Theorem: Let $\{V,R\}$ be a vectoid and o its neutral element, $\{V,\leqq\}$ a complete lattice, and $\{T,\leqq\}$ a symmetric screen (S1), (S2), (S) (respectively a symmetric lower screen, respectively a symmetric upper screen), $\square: V \to T$ an antisymmetric rounding (R1), (R3), and S a screen-ringoid of R. In T let an operation $\boxplus: T \times T \to T$ and a multiplication $\boxdot: S \times T \to T$ be defined by formula (R). Then

1) $\{T,S\}$ is also a vectoid with neutral element o and

$$\text{(RG1)} \quad \bigwedge_{a,b\in T} (a + b \in T \Rightarrow a \boxplus b = a + b) \wedge \bigwedge_{a\in S} \bigwedge_{a\in T} (a \cdot a \in T \Rightarrow a \boxdot a = a \cdot a)$$

$$\text{(RG3)} \quad \bigwedge_{a\in T} \square a = -a.$$

2) If $\square: V \to T$ is monotone (R2) $\Rightarrow$

$$\text{(RG2)} \quad \bigwedge_{a,b,c,d\in T} (a + b \leqq c + d \Rightarrow a \boxplus b \leqq c \boxplus d) \quad \bigwedge_{a,b\in S} \bigwedge_{a,b\in T} (a \cdot a \leqq b \cdot b \Rightarrow a \boxdot a \leqq b \boxdot b).$$

3) If $\square: V \to T$ is downwardly, respectively upwardly, directed (R4) $\Rightarrow$

$$\text{(RG4)} \quad \bigwedge_{a,b\in T} a \boxplus b \leqq a + b,$$

$$\text{respectively} \quad \bigwedge_{a,b\in T} a + b \leqq a \boxplus b,$$

$$\bigwedge_{a\in S} \bigwedge_{a\in T} a \boxdot a \leqq a \cdot a,$$

$$\text{respectively} \quad \bigwedge_{a\in S} \bigwedge_{a\in T} a \cdot a \leqq a \boxdot a.$$

4) If $\{V,R,\leqq\}$ is weakly ordered (OV1),(OV2) and $\square: V \to T$ monotone $\Rightarrow \{T,S,\leqq\}$ is weakly ordered, i.e., (OD1),(OD2) hold.

5) If $\{V,R,\leqq\}$ is ordered (OV3) and $\square: V \to T$ monotone $\Rightarrow \{T,S,\leqq\}$ is ordered, i.e., (OV3) holds. $\square$

Theorem: Let $\{V,R\}$ be a multiplicative vectoid with neutral elements o and e, $\{V,\leqq\}$ a complete lattice and $\{T,\leqq\}$ a symmetric screen (respectively a symmetric lower

screen, respectively a symmetric upper screen), $\square: V \to T$ an antisymmetric rounding and S a screen ringoid of R. In T let operations $\circledast: T \times T \to T$, $* \in \{+, \cdot\}$ and a multiplication $\boxdot: S \times T \to T$ be defined by formula (R). Then

1) $\{T,S\}$ is a multiplicative vectoid with neutral elements $\boldsymbol{o}$ and $\boldsymbol{e}$ and (RG1) holds for all operations as well as (RG3).

2) If $\square: V \to T$ is monotone $\Rightarrow$ (RG2) for all operations.

3) If $\square: V \to T$ is downwardly, respectively upwardly, directed $\Rightarrow$ (RG4) for all operations.

4) If $\{V,R,\leqq\}$ is weakly ordered and $\square: V \to T$ monotone $\Rightarrow \{T,S,\leqq\}$ is a weakly ordered multiplicative vectoid.

5) If $\{T,S,\leqq\}$ is an ordered multiplicative vectoid and $\square: V \to T$ monotone $\Rightarrow \{T,S,\leqq\}$ is also an ordered multiplicative vectoid. □

All statements of these theorems are easily verified. The proofs show that our assumptions (S1),(S2),(S), (R1),(R2),(R3),(R), respectively (R4) are really necessary in order to get the desired structure in T. If we change these properties or do not realize them strictly we get a different structure in the subset T.

The last two theorems show that the structure of a weakly ordered or ordered vectoid is invariant with respect to monotone and antisymmetric roundings into a symmetric screen if the operations in the subset are defined by formula (R). This describes all structures in Fig. 1 in the lines 2, 3, 5, 6, 8, 9, 11 and 12.

A few words still have to be said about the interval structures. This section is the most interesting one of the whole theory. It cannot, however, be treated within this paper. See [12], [13]. In every interval set listed in Fig. 1 we have two order relations. With respect to $\leqq$ the structures are ordered, respectively weakly ordered, in the complex case and the rounding is monotone. This guarantees that finally we will get the same structure on the upper screen. The other order relation is the inclusion $\subseteq$ with respect to which the upper screens are defined. The rounding is antisymmetric, monotone, and upwardly directed with respect to the inclusion.

Further with respect to the inclusion all operations are monotone, i.e., the property

$$\bigwedge_{A,B,C,D} (A \subseteq B \wedge C \subseteq D \Rightarrow A * C \subseteq B * D)$$

is valid for all operations $* \in (\{+,-,\cdot,/\}$ and not only for the addition.

At the first view some of our interval spaces in Fig. 1 seem to be unrealistic. Actual interval computations are not done in the set of intervals of vectors or matrices *IVR, IMR,* respectively *IVC, IMC,* but in the sets of vectors and matrices with interval components *VIR, MIR,* respectively *VIC, MIC.* It can, however, be shown by not at all trivial theorems that the spaces *IVR* and *VIR, IMR* and *MIR, IVC* and *VIC, IMC* and *MIC* are isomorphic with respect to the algebraic structure and the order relation $\leqq$. See [13]. This finally shows that the structures which we have derived also in the interval cases are realistic.

REFERENCES

[1] N. Apostolatos, H. Christ, H. Santo, and H. Wippermann, "Rounding control and the algorithmic language ALGOL-68," Universität Karlsruhe, Rechenzentrum, rep., pp. 1–9, July 1968.

[2] H. Christ, "Realisierung einer Maschinenintervallarithmetik auf beliebigen ALGOL-60 Compilern," *Elektronische Rechenanlagen,* vol. 10, no. 5, pp. 217–222, 1968.

[3] G. Bohlender, "Floating-point computation of functions with maximum accuracy," see this Symposium.

[4] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1967.

[5] K. Grüner, "Fehlerschranken für lineare Gleichungssysteme, 1975," *Computing,* to be published.

[6] H. C. Haas, "Implementierung der komplexen Gleitkommaarithmetik mit maximaler Genauigkeit," Diplomarbeit, Institut für Angewandte Mathematik, Universität Karlsruhe, pp. 1–118, 1975.

[7] J. Herzberger, "Metrische Eigenschaften von Mengensystemen und einige Anwendungen," Dr.-Dissertation, Universität Karlsruhe, pp. 1–49, 1969.

[8] D. Knuth, *The Art of Computer Programming,* Vol. 2. Reading, MA: Addison-Wesley, 1969.

[9] U. Kulisch, "An axiomatic approach to rounded computations," Mathematics Research Center, The University of Wisconsin, Madison, WI, Tech. Summary, Rep. 1020, pp. 1–29, Nov. 1969; and *Num. Math.,* 18, pp. 1–17, 1971.

[10] ——, "On the concept of a screen," Mathematics Research Center, The Unversity of Wisconsin, Madison, WI, Tech. Summary Rep. 1084, pp. 1–12, July 1970; and *ZAMM,* vol. 53, pp. 115–119, 1973.

[11] ——, "Rounding invariant structures," Mathematics Research Center, The University of Wisconsin, Madison, WI, Tech. Summary Rep. 1103, pp. 1–47, Sept. 1970.

[12] ——, "Interval arithmetic over completely ordered ringoids," The University of Wisconsin, Madison, WI, Tech. Summary Rep. 1105, pp. 1–56, Sept. 1970.

[13] ——, "Grundlagen des Numerischen Rechnens, Niederschrift einer Vorlesung, gehalten im WS 1970/1971," Universität Karlsruhe, pp. 1–250.

[14] ——, "Implementation and formalization of floating-point arithmetics," IBM T. J. Watson Research Center, Rep. R.C. 4608, pp. 1–50, Nov. 1973; and *Computing,* vol. 14, pp. 323–348, 1975.

[15] ——, "Über die Arithmetik von Rechenanlagen," *Jahrbuch Überlicke Mathematik 1975,* Wissenschaftsverlag des Bibliographischen Instituts Mannheim/Wien/Zürich, pp. 68–108.

[16] U. Kulisch and G. Bohlender, "Formalization and implementation of floating-point matrix operations," Universität Karlsruhe, Rep., pp. 1–35, Sept. 1974; and *Computing,* vol. 16, pp. 239–261, 1976.

[17] B. Lortz, "Eine Langzahlarithmetik mit optimaler einseitiger Rundung," Dr.-Dissertation, Universität Karlsruhe, pp. 1–48, 1971.

[18] H. Rutishauser, "Verusch einer Axiomatik des Numerischen Rechnens," Kurzvortrag, GAMM-Tagung, Aachen, 1969.

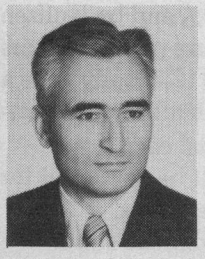
[19] C. Ullrich, "Rundungsinvariante Strükturen mit äußeren Verknüpfungen," Dr.-Dissertation, Universität Karlsruhe, pp. 1–67, 1972.

[20] C. Ullrich, "Über die beim numerischen Rechnen mit komplexen Zahlen und Intervallen vorliegenden mathematischen Strukturen," *Computing,* vol. 14, pp. 51–65, 1975.

[21] J. H. Wilkinson, *Rundungsfehler.* Berlin: Springer-Verlag, 1969.

[22] J. M. Yohé, "Roundings in floating-point arithmetic," *IEEE Trans. Comput.,* vol. C-22, pp. 577–586, June 1973.

[23] U. Kulisch, *Grundlagen des Numerischen Rechnens—Mathematische Begrundung der Rechnerarithmetik.* Mannheim: Bibliographisches Institut, 1976.



Ulrich Kulisch studied mathematics and physics at the Technical University, Munich, Germany, and the University of Munich, Munich, Germany, from 1953 to 1958. He received the Dr. rer. nat. degree in mathematics in 1961 and the Habilitation degree in mathematics in 1963, both from the Technical University of Munich.

Since 1963 he has taught mathematics at the Technical University of Munich, the University of Munich, and the University of Karlsruhe, Karlsruhe, Germany. Since 1967 he has been Full Professor of Mathematics and Director of the Institute of Applied Mathematics at the University of Karlsruhe. From 1966 to 1970 he was also Director of the Computer Center of this University. During the years 1969 and 1970 he was on academic leave with the Mathematics Research Center, The University of Wisconsin, Madison and 1972 to 1973 at the IBM T. J. Watson Research Center, Yorktown Heights, NY. He has published about 30 research articles in mathematics and computer sciences and two books, the first one in 1969 together with J. Heinhold on *Analog and Hybrid Computations* and the second one in 1976 about *Fundamentals of Numerical Computations—Mathematical Foundation of Computer Arithmetic*. Since 1968 he has been editor of the book series "Reihe Informatik" and since 1974 also of the series "Jahrbuch Überlicke Mathematik" by Bibliographisches Institut, Mannheim, West Germany.

Floating-Point Computation of Functions with Maximum Accuracy

GERD BOHLENDER

***Abstract*—Algorithms are given that compute multiple sums and products and arbitrary roots of floating-point numbers with maximum accuracy. The summation algorithm can be applied to compute scalar products, matrix products, etc. For all these functions, simple error formulas and the smallest floating-point intervals containing the exact result can be obtained.**

***Index Terms*—Accuracy, errors, floating-point computations, multiple-length mantissas, roots of floating-point numbers, rounding.**

I. Introduction

OUR AIM is to approximate functions[1] $f: \mathbf{R}^n \to \mathbf{R}^p$ on a floating-point system T. For $b,l \in \mathbf{N}$, $b \geqq 2$, $l \geqq 1$, the floating-point system $T_{b,l}$ with base b and l-digit mantissa is defined by

$$T_{b,l} := \{0\} \cup \{x = *m \cdot b^e;\ * \in \{+,-\},\ m = 0 \cdot m[1] \cdots m[l],\ m[i] \in \{0,1,\cdots,b-1\},\ m[1] \neq 0,\ e \in \mathbf{Z}\}. \quad (1)$$

x is then called a floating-point number with sign $* = \operatorname{sgn}(x)$, mantissa $m = \operatorname{mant}(x)$, and exponent $e = \exp(x)$. As the base b will be kept fixed throughout the paper, we will suppress the index b and write shortly T_l or T. For the present, we do not consider the finite exponent range that is available in practice, as this would necessitate complicated exponent overflow and underflow discussions. Instead, we give remarks on the influence of limiting the exponent range on our algorithms.

The best possible approximation for $f(x)$ is $\square f(x)$, wherein $\square: \mathbf{R}^p \to T^p$ denotes a rounding.[2] We will restrict ourselves here to the roundings ∇, Δ and $\square_\mu$ ($\mu = 0(1)b$). For $p = 1$ these roundings are defined as follows:

$$\bigwedge_{x \in \mathbf{R}} \nabla x := \max\{y \in T; y \leqq x\} \quad (2)$$

$$\bigwedge_{x \in \mathbf{R}} \Delta x := \min\{y \in T; x \leqq y\} = -\nabla(-x) \quad (3)$$

$$\bigwedge_{x \geqq 0} \square_b x := \nabla x \wedge \bigwedge_{x < 0} \square_b x := \Delta x \quad (4)$$

$$\bigwedge_{x \geqq 0} \square_0 x := \Delta x \wedge \bigwedge_{x < 0} \square_0 x := \nabla x \quad (5)$$

and for $\mu = 1(1)b - 1$

$$\bigwedge_{x \geqq 0} \square_\mu x: \begin{cases} \nabla x & \text{for } x \in [\nabla x, S_\mu(x)) \\ \Delta x & \text{for } x \in [S_\mu(x), \Delta x] \end{cases}$$

$$\bigwedge_{x < 0} \square_\mu x := -\square_\mu(-x), \quad (6)$$

Manuscript received January 20, 1976; revised October 15, 1976.

The author is with the Institute of Applied Mathematics, University of Karlsruhe, Karlsruhe, Germany.

[1] **N, Z,** and **R** denote the sets of nonnegative integers, integers and reals, respectively. For any given set S, S^p denotes the set of p-tuples with components out of S. $\{x_i; P(x)\}$ denotes the set of all elements x with property $P(x)$.

[2] As regards general definitions, we refer to Kulisch [5].