Mathematical Foundation of Computer Arithmetic

U. Kulisch

During the last 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 Some of these papers are still unpublished.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 in dependence of 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.

1. 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 of these spaces occasionally also the corresponding complex spaces C,VC and MC occur. A couple of 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 $\{M,\leq\}$ by IM we get the spaces IR,IVR,IMR and IC,IVC and IMC. See the second column in figure 1.

Since a real number in general is represented by an infinite b-adic expansion the algorithms given in these spaces in general can not be executed within them. The real numbers, therefore, get approximated by a subset T in which all operations are simple and fast performable. On computers for T a floating-point system with a finite number of digits in the mantissa is used. If the desired accuracy can not 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 column in figure 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 the table of figure 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 column of figure 1 operations are to be defined. See the fifth column in figure 1. Furthermore the lines in figure 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 column in figure 1 should be available possibly as operators for special data types.

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

Figure 1: Table of the spaces and operations occuring in numerical computations

The following treatment is devoted to the question 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 to describe all the sets listed in figure 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. (Definitions see below).

We are now going to describe this general principle a little more closely. Let M be one of the sets listed in figure 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 figure 1 the structure is well known in the sets of R,VR,MR,C,VC and MC. Let now M be one of these sets and $*$ one of the operations defined in M. Then also in the powerset PM, which is the set of all subsets of M, an operation $*$ can be defined by

$$\bigwedge_{A,B \in PM} A * B := \{a * b \mid a \in A \wedge b \in B\} \quad (1)$$

If we apply this definition for all operations $*$ of M we shall see below that also in the powerset a structure $\{PM,\overline{PM}\}$ can be derived from that in $\{M,\overline{M}\}$. Summarizing this result we can say that in figure 1 the structure $\{M,\overline{M}\}$ is known always 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 transfered 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

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

Further in all structures of figure 1 which we already know a minusoperator is defined and if for instance S and T are floating-point systems it is easy to see (see [11],[12],[13],[14],[16],[19]) that in every line in figure 1 all subsets have the property

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

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

It will turn out below that the rounding $\square : M \to N$ is not only responsible 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 figure 1 we get a full generalization $\overline{N} \subset \overline{M}$. In the next an possible further steps $\overline{N} = \overline{M}$.

Let us now consider the question 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 attempted to try it with useful mapping properties like isomorphism and homomorphism. But it is easy to see that an isomorphism can not be achieved and it can be shown by simple examples in the case of the first line of figure 1 that also an homomorphism can not 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 go as far to an homomorphism as possible. Let us therefore at first repeat the definition of an homomorphism.

<u>Definition:</u> 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

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

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

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

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

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

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

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

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

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

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

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

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 figure 1 is a difficult task which can not be solved within this paper. It is, however, an essential result that it can be given in all cases. (See [11], [12],[13],[14],[16],[19],[20]).

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

2. 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

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

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 is of the magnitude 10^{50} and b of the magnitude 10^{-50} for the representation of $a + b$ about 100 decimal digits in the mantissa would be necessary . 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 respresentable value $a \tilde{*} b$ with the property $\square(a * b) = \square(a \tilde{*} b)$. Then $a \tilde{*} b$ can be used to define $a \boxed{*} b$ by

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

The proof of this assertion has to be given by concrete algorithms in all cases of figure 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 figure 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:

∇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

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

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

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

$$\bigwedge_{\beta^{e1-1} \leqq a \leqq B} \square_\mu^{1)} a = \begin{cases} \nabla a & \text{for } a \in [\nabla a, S_\mu(a)) \\ \Delta a & \text{for } a \in [S_\mu(a), \Delta a] \end{cases} \qquad (2)$$

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

where $B := 0.(\beta-1)(\beta-1)\ldots(\beta-1) \cdot \beta^{e2}$ 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 $\{\nabla, \Delta, \square_\mu, \ \mu = 0(1)\beta\}$ are not independent of each other. The following relations are easily verified:

$$\Delta a = -\nabla(-a) \qquad (3)$$

$$\nabla a = -\Delta(-a) \qquad (4)$$

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

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

All roundings $\square_\mu : R \to T, \mu = 0(1)\beta$, are further antisymmetric functions. From (1),(2),(3),(4) follows that all these roundings can be expressed by the monotone downwardly (resp. 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 $a \boxed{*} b = \square(a * b) = \square(a \tilde{*} b)$. (R).
5. Composition, i.e. combination of the resulting exponent part and mantissa to a floating-point

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

number. (C).

Figure 2 gives a graphical diagram of these five steps. A more detailed discussion of these steps can be found in the literature [13],[14],[15],[17],[22],[8].

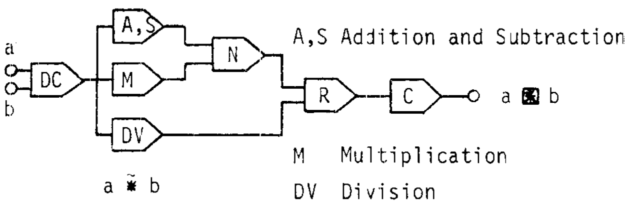


Figure 2: Flow diagram for the arithmetic operations

The algorithms can be implemented using accumulators of different length. 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 figure 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 figure 3. This algorithm shows that a further reduction of the length of the accumulator is impossible if formula (R) strictly is to be realized.

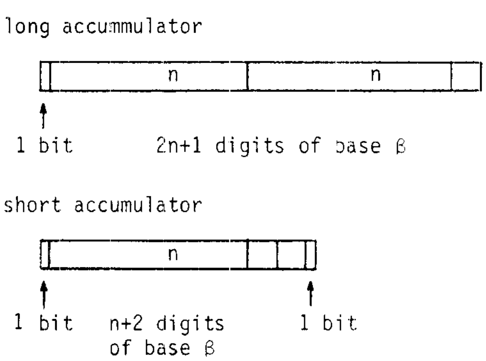


Figure 3: Long and short accumulators

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 $\{\nabla,\Delta,\square_\mu,\ \mu=o(1)\beta\}$ formula (R) holds.

With these algorithms the question of implementation is solved in case of the first line of figure 1 and since formula (R) has also been implemented with the roundings ∇ and Δ also for most of the interval lines in figure 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 also $\square : MR \to MT$ is a rounding. If further the rounding $\square : R \to T$ is monotone, directed, antisymmetric respectively then also the rounding $\square : MR \to MT$ is monotone, directed, antisymmetric respectively.

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

$$\text{(R)} \quad \bigwedge_{A,B\in MT} A \boxed{*} B := \square(A * B) \text{ for all } * \in \{+,\cdot\}.$$

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

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

Therein 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

$$A \boxdot B := \square(A \cdot B) := \square\Big(\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}\Big) \tag{5}$$

where in

$$\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j} \tag{6}$$

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 (6) is exactly representable. The algorithms show, however, that whenever (6) is not representable it can be replaced by an appropriate and representable value

$$\widetilde{\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}} \tag{7}$$

with the property

$$\square(A \cdot B) = \square\Big(\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}\Big) = \square(A \tilde{\cdot} B) = \square\Big(\widetilde{\sum_{\nu=1}^{r} a_{i\nu} b_{\nu j}}\Big) \tag{8}$$

Then (8) can be used to define (5). The explicit algorithms prove this assertion. See [16],[3].

In order to realize (8) 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 (8) can be generated if the sum

$$z := \square\Big(\sum_{i=1}^{r} x_i\Big) = \square\Big(\widetilde{\sum_{i=1}^{r} x_i}\Big) \tag{9}$$

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 (9) of $n,n+1,\ldots,L=2n$ correct digits just by rounding the intermediate result $\widetilde{\sum_{i=1}^{r} x_i}$ to other length. These algorithms again can be separated into several independent steps which means that the intermediate result $\widetilde{\sum_{i=1}^{r} x_i}$ can be chosen independently of the rounding function such that for all roundings of the set $\square$ $\{\nabla,\Delta,\square_\mu,\ \mu=o(1)\beta\}$ the equality

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

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 figure 4.

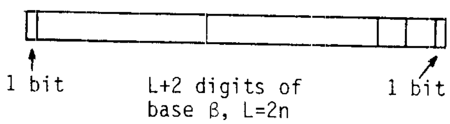


Figure 4: Length of the accumulator for scalar prodructs

With this algorithm the question of implementation is solved not only in case of the third line of figure 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 been realized also for the roundings ∇ and Δ in all cases of interval structures occuring in figure 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

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

has to be realized. But also this problem has been solved in [6]. In this case still a little longer accumulator is necessary. The running time for a software solution of this 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 in 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 figure 1. Figure 5a describes the way how 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.

a)
$$\begin{tikzcd} R \arrow[r] \arrow[d] & S \arrow[r] \arrow[d] & T \arrow[d] \\ MR & MS & MT \end{tikzcd}$$

b)
$$\begin{tikzcd} R \arrow[r] \arrow[d] & S \arrow[r] & T \\ MR \arrow[r] & MS \arrow[r] & MT \end{tikzcd}$$

Figure 5: Definition of floating-point matrix operations

Figure 5b 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

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

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

(RG3) $\bigwedge_{A\in MT} -A = \boxed{-} A := (-E) \boxed{\cdot} A$, E unit matrix

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

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

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

In this case the $\leqq$ sign means the inclusion and (RG4) then says that the result of an operation inthe 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 $\square : R \rightarrow T$ a monotone rounding and let $\delta(\square a) := a - \square a$ denote the absolute rounding error and $\varepsilon := \delta(\square a)/a$ the relative rounding error. Then

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

where

$$\varepsilon^{*} := \begin{cases} \frac{1}{2}\, \beta^{1-n} & \text{for the rounding to the nearest floating-point number} \\ \beta^{1-n} & \text{else} \end{cases} \tag{10}$$

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

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

where ε^{*} is defined by (10).

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 <u>bounds</u> if formula (R) is strictly implemented.

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

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

Then

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

where ε^{*} is defined by (10) 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 $* \in \{+,-,\cdot\}$:

$$\bigwedge_{A,B \in MT} (\bigwedge_{i,j} b^{e1-1} \leqq z_{ij} < b^{e2} \Rightarrow A \boxed{*} B = (z_{ij}(1-\varepsilon_{ij})) \text{ with } |\varepsilon_{ij}| < \varepsilon^{*} \Rightarrow |A * B - A \boxed{*} B| \leqq \varepsilon^{*} \cdot |A * B|). \tag{11}$$

This is the same simple formula with the same ε^{*} which we have got 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 figure 5a. Furthermore (11) is more accurate. In [5] an error analysis of the Gauß algorithm for linear equations using formula (11) is given. See also [4].

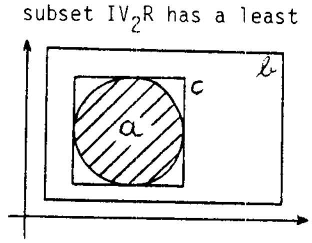
In contrary to most error estimations in Numerical Mathematics the error formulas derived in this paper lead to absolute error <u>bounds</u> if formula (R) is strictly implemented.

3. <u>The Structure of Computer Arithmetic</u>

In literature several attempts to formulize computer arithmetic are known. All these models are only interested to describe the relationship between the real numbers and a floating-point system. It turns out, however, that the real numbers have to many very special properties in order to recognize all essential properties already at this model. Only the entirety of structures listed in figure 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 powerset of an ordered algebraic structure and its intervals there exists mathematically the same relationship than 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 figure 1. All these sets are ordered with respect to certain order relations. Let us consider the interval vectors of dimension 2, IV_2R. This 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 powerset 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. Figure 6.
2. For all $a \in PV_2R$ the set of all upper bounds in the subset IV_2R has a least element. Figure 6.



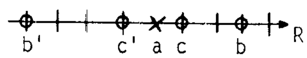
$a \in PV_2R$
$b,c \in IV_2R$
$a \subseteq b \wedge a \subseteq c \wedge$
$\wedge \bigwedge_{b \in IV_2R} (a \subseteq b \Rightarrow c \subseteq b)$

<u>Figure 6:</u> To the concept of a screen

These two properties also characterize the relationship between any set of figure 1 and its subset(s) on the right hand side. Let us now consider the set of real numbers R 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 $\leq$ of the real numbers) in the subset T. Figure 7.
2. For all $a \in R$ the set of all upper bounds in the subset T has a least element. Figure 7.

In this case corresponding properties are also valid for the lower bounds.



$$a \in R$$
$$b, c \in T$$
$$a \leq b \wedge a \leq c \wedge$$
$$\wedge \bigwedge_{b \in T} (a \leq b \Rightarrow c \leq b)$$

Figure 7: To the concept of a screen

We summarize these properties by the

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

(S1) $\bigwedge_{a \in M} L(a) \cap T \neq \emptyset$ resp. $\bigwedge_{a \in M} U(a) \cap T \neq \emptyset$

(S2) $\bigwedge_{a \in M} \bigvee_{x \in L(a) \cap T} \bigwedge_{b \in L(a) \cap T} b \leq x$ resp.

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

If $T \subseteq M$ is sumultaneously 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 nesessary and sufficient conditions can be derived. See [10].

Let $\{M, \leq\}$ 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

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

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

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

(S1') $o(M) = o(T)$ (resp. $i(M) = i(T)$) and

(S2') $\{T, \leq\}$ is a complete supremum-subformation (resp. a complete infimum-subforation) of $\{M, \leq\}$.

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

For the proof see [10]. Now it can be shown that all sets in figure 1 are screens of the set(s) on their left hand side. See [11],[12],[13].

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

$$\bigwedge_{a \in M} \nabla a = \sup(L(a) \cap T) \quad \text{resp.} \quad \bigwedge_{a \in M} \Delta a = \inf(U(a) \cap T).$$

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

$$\bigwedge_{a,b \in T} a \boxtimes b = \nabla(a * b) \quad \text{resp.} \quad \bigwedge_{a,b \in T} a \boxtimes b = \Delta(a * b).$$

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

We are now going to define the special structures of a weakly ordered resp. 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 figure 1.

Definition: A non empty 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 $/ : R \times R \setminus N \longrightarrow R$

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 of (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 $\{R,\leq\}$ is an ordered [1] set and

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

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

A weakly ordered ringoid resp. divisionsringoid is called an ordered ringoid resp. an ordered divisionsringoid if

(OD3) $\bigwedge_{a,b,c \in R} (o \leq a \leq b \wedge c \geq o \Rightarrow a \cdot c \leq b \cdot c \wedge c \cdot a \leq c \cdot b)$

respectively

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

The uniqueness of x can be used for the following

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

$$\bigwedge_{a \in R} -a := x \cdot a \tag{1}$$

$$\bigwedge_{a,b \in R} a - b := a + (-b). \quad \square \tag{2}$$

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) $\Rightarrow (a \leq b \Rightarrow -b \leq -a)$

In general there do not exist inverse elements of the addition 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 divisionringoid we get further

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

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

In a weakly ordered ringoid holds

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

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

In an ordered ringoid respectively ordered division ringoid we get

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

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

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

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

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

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

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 than in the real number field. In an ordered ringoid for all elements which are comparable with o with respect to $\leq$ and $\geq$ the same rules for inequalities hold than in the real number field.

Applications: Let R be a divisionringoid.

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 also MR is a ringoid.

If PR denotes the powerset of R and in PR operations are defined by 1. formula (1) then also PR becomes a divisionringoid.

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 than in the complex number field then also CR becomes a divisionringoid.

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

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

For the proofs of these results see [11] and [13].

1) $\{R,\leq\}$ is an ordered set means $\leq$ is a reflexive (O1) transitive (O2) and antisymmetric (O3) relation

If in figure 1 R is an ordered divisionringoid 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) (resp. a symmetric lower screen resp. a symmetric upper screen), $\square: R \rightarrow T$ an antisymmetric rounding (R1),(R3) and let in T operations $\boxed{*}$, $* \in \{+,\cdot\}$, be defined by formula (R). Then

(A) in T the following properties hold: (D1),(D2) for o, (D3) for e, (D4),(D5) for -e and

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

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

(B) if $\square: R \rightarrow T$ is monotone (R2) $\Rightarrow$

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

(C) if $\square: R \rightarrow T$ is upwardly resp. downwardly directed (R4) $\Rightarrow$

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

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

(D) if R is weakly ordered (OD1),(OD2) and $\square: R \rightarrow T$ montone $\Rightarrow$
T is weakly ordered, i.e. (OD1), (OD2) hold.

(E) if R is ordered (OD3) and $\square: R \rightarrow T$ monotone $\Rightarrow$ in T (OD3) holds. □

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

(A) 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)

(B) if $\square: R \rightarrow T$ is monotone $\Rightarrow$ (RG2) for $* \in \{+,-,\cdot,/\}$

(C) if $\square: R \rightarrow T$ is downwardly resp. upwardly directed $\Rightarrow$ (RG4) for $* \in \{+,-,\cdot,/\}$

(D) if R is an ordered divisionringoid and $\square: R \rightarrow T$ monotone $\Rightarrow$ in T (OD4) holds. □

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

(D5c): $(-e) \boxdot a \underset{(R)}{=} \square(-a) \underset{(R3)}{=} -\square a \underset{(R1)}{=} -a \underset{(S)}{\in} T \qquad (3)$

$$(-e) \boxdot (a \boxplus b) \underset{(R)}{=} \square((-e) \cdot \square(a+b)) \underset{(R3)}{=}$$
$$= \square(\square(-(a+b))) \underset{(R1)}{=} \square(-(a+b)) \underset{(D5c)_R}{=}$$
$$= \square((-a)+(-b)) \underset{(R)}{=} (-a) \boxplus (-b) \underset{(3)}{=}$$
$$= ((-e) \boxdot a) \boxplus ((-e) \boxdot b).$$

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

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 propertiy which can not be proved by a general theorem is (D6). The proof of this property is a difficult task in all cases of figure 1. Concerning to these proofs we refer to the literature [11], [12],[13],[14],[16],[20].

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

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

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 figure 1) is that of a linearly ordered divisionringoid.

We are now going to define the structure of the "higher dimensional spaces" listed in figure 1. We shall later see that the structure of a weakly ordered resp. an ordered vectoid under the assumptions (S1),(S2),(S),(R1),(R2),(R3) and (R) describes the structures in the lines 2,3,5,6,8,9,11,12 of figure 1.

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

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

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

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

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

fulfils the following properties:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Definition: In a vectoid we define a subtraction by

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

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 holds

(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

(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) $a \leqq b \leqq o \wedge o \leqq c \leqq d \Rightarrow ad \leqq bc \leqq o$
(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) $a \leqq b \leqq o \wedge c \leqq d \leqq o \Rightarrow o \leqq bd \leqq ac \wedge o \leqq db \leqq ca.$ □

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 than 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 than in the real vector space.

Applications:

Let {V,R} be a vectoid. Then the powerset {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 $VR := R \times R \times \ldots \times R$ 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-tupels 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 $\{VR,R,\leq\}$ as well as $\{VR,MR,\leq\}$ are weakly ordered respectively ordered vectoids.
$\{MR,R,\leq\}$ is a weakly ordered respectively an ordered multiplicative vectoid.
The proof of these results is left to the reader. See [19] and [13].

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

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

<u>Theorem:</u> Let {V,R} be a vectoid and o its neutral element, $\{V,\leq\}$ a complete lattice and $\{T,\leq\}$ a symmetric screen (S1),(S2),(S) (resp. a symmetric lower screen resp. a symmetric upper screen), $\square: V \to T$ an antisymmetric rounding (R1),(R3) and S a screenringoid 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

(A) {T,S} is also a vectoid with neutral element o and

(RG1) $\bigwedge_{a,b\in T} (a + b\in T \Rightarrow a \boxplus b = a + b) \wedge$

$\bigwedge_{a\in S} \bigwedge_{a\in T} (a \cdot a\in T \Rightarrow a \boxdot a = a \cdot a)$

(RG3) $\bigwedge_{a\in T} \boxminus a = -a$,

(B) if $\square: V \to T$ is monotone (R2) $\Rightarrow$

(RG2) $\bigwedge_{a,b,c,d\in T} (a + b \leq c + d \Rightarrow a \boxplus b \leq c \boxplus d)$

$\bigwedge_{a,b\in S} \bigwedge_{a,b\in T} (a \cdot a \leq b \cdot b \Rightarrow a \boxdot a \leq b \boxdot b)$

(C) if $\square: V \to T$ is downwardly resp. upwardly directed (R4) $\Rightarrow$

(RG4) $\bigwedge_{a,b\in T} a \boxplus b \leq a + b$ resp. $\bigwedge_{a,b\in T} a+b \leq a \boxplus b$

$\bigwedge_{a\in S} \bigwedge_{a\in T} a \boxdot a \leq a\cdot a$ resp. $\bigwedge_{a\in S} \bigwedge_{a\in T} a\cdot a \leq a \boxdot a$

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

(E) if $\{V,R,\leq\}$ is ordered (OV3) and $\square : V \to T$ monotone $\Rightarrow \{T,S,\leq\}$ is ordered, i.e. (OV3) holds. □

<u>Theorem:</u> Let {V,R} be a multiplicative vectoid with neutral elements o and e, $\{V,\leq\}$ a complete lattice and $\{T,\leq\}$ a symmetric screen (resp. a symmetric lower screen resp. a symmetric upper screen), $\square : V \to T$ an antisymmetric rounding and S a screenringoid 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

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

(B) if $\square: V \to T$ is monotone $\Rightarrow$(RG2) for all operations

(C) if $\square: V \to T$ is downwardly resp. upwardly directed $\Rightarrow$(RG4) for all operations

(D) if $\{V,R,\leq\}$ is weakly ordered and $\square : V \to T$ monotone $\Rightarrow \{T,S,\leq\}$ is a weakly ordered multiplicative vectoid

(E) if $\{T,S,\leq\}$ is an ordered multiplicative vectoid and $\square : V \to T$ monotone $\Rightarrow \{T,S,\leq\}$ 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) resp. (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 figure 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 chapter is the most interesting one of the whole theory. It can, however, not be treated within this paper. See [12],[13]. In every interval set listed in figure 1 we have two order relations. With respect to $\leq$ the structures are ordered respectively weakly ordered in the complex case and the rounding is monotone. This guarantees it finally that we 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 figure 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 $\leq$. See [13]. This finally shows that the structures which we have derived also in the interval cases are realistic.

References

[1] Apostolatos, N., Christ, H., Santo, H., Wippermann, H.: Rounding Control and the Algorithmic Language ALGOL-68 . Report, Universität Karlsruhe, Rechenzentrum, Juli 1968, S. 1-9

[2] Christ, H.: Realisierung einer Maschinenintervallarithmetik auf beliebigen ALGOL-60 Compilern. Elektronische Rechenanlagen, 10 (1968), H.5, S. 217-222.

[3] Bohlender, G.: Floating-Point Computation of Functions with Maximum Accuracy. See this Symposium

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

[5] Grüner, K.: Fehlerschranken für lineare Gleichungssysteme, 1975. To appear in Computing.

[6] Haas, H.Chr.: Implementierung der komplexen Gleitkommaarithmetik mit maximaler Genauigkeit. Diplomarbeit, Institut für Angewandte Mathematik, Universität Karlsruhe, S. 1-118, 1975

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

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

[9] Kulisch, U.: An Axiomatic Approach to Rounded Computations, Mathematics Research Center, The University of Wisconsin, Madison, Wisconsin, Technical Summary Report Nr. 1020, Nov. 1969, p. 1-29, and Num. Math. 18, p. 1-17 (1971)

[10] - - - - - : On the Concept of a Screen, Mathematics Research Center, The University of Wisconsin, Madison, Wisconsin, Technical Summary Report Nr. 1084, July 1970, p. 1-12, and ZAMM 53, 115-119 (1973)

[11] - - - - - : Rounding Invariant Structures, Mathematics Research Center, The University of Wisconsin, Madison, Wisconsin, Technical Summary Report Nr. 1103, Sept. 1970, p.1-47

[12] - - - - - : Interval Arithmetic over Completely Ordered Ringoids, The University of Wisconsin, Madison, Wisconsin, Technical Summary Report Nr. 1105, Sept. 1970, p. 1-56

[13] - - - - - : Grundlagen des Numerischen Rechnens, Niederschrift einer Vorlesung, gehalten im WS. 1970/71 an der Universität Karlsruhe, S. 1-250

[14] - - - - - : Implementation and Formalization of Floating-Point Arithmetics, IBM T.J. Watson Research Center, Report Nr. R.C. 4608, Nov. 1973, p. 1-50, to appear in part in Computing under the title: Formalization and Implementation of Floating-Point Arithmetics

[15] - - - - - : Über die Arithmetik von Rechenanlagen, Jahrbuch Überblicke Mathematik 1975, Wissenschaftsverlag des Bibliographischen Instituts Mannheim/Wien/Zürich, S. 68-108

[16] Kulisch, U. and Bohlender, G.: Formalization and Implementation of Floating-Point Matrix Operations, Report, Universität Karlsruhe, Sept. 1974, p. 1-35, to appear in Computing

[17] Lortz, B.: Eine Langzahlarithmetik mit optimaler einseitiger Rundung, Dr.-Dissertation, Universität Karlsruhe, 1971, S. 1-148

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

[19] Ullrich, Chr.: Rundungsinvariante Strukturen mit äußeren Verknüpfungen. Dr.-Dissertation, Universität Karlsruhe, 1972, S. 1-67

[20] Ullrich, Chr.: Über die beim numerischen Rechnen mit komplexen Zahlen und Intervallen vorliegenden mathematischen Strukturen. Computing 14, S. 51-65 (1975)

[21] Wilkinson, J.H.: Rundungsfehler, Springer-Verlag, Berlin, Heidelberg, New York 1969

[22] Yohê, J.M.: Roundings in Floating-Point Arithmetic, JEEE Transactions on Computers, Vol. C-22, Nov. 6, June 1973, p. 577-586

Address:

Prof. Dr. Ulrich Kulisch
Institut für Angewandte Mathematik
Universität Karlsruhe

D-75 Karlsruhe
Kaiserstr. 12
West Germany