Computing 16, 239—261 (1976)

Formalization and Implementation of Floating-Point Matrix Operations*

U. Kulisch and **G. Bohlender**, Karlsruhe

Received December 17, 1974

Abstract — Zusammenfassung

Formalization and Implementation of Floating-Point Matrix Operations. The paper shows that floating-point matrix operations can be implemented in a way which leads to reasonable mathematical structures as well as to sensible compatibility properties between these structures and the structure of the real matrices. It turns out, for instance, that all the rules of the minus-operator for real matrices can be saved and that for all elements which are comparable with 0 with respect to $\leqq$ and $\geqq$ the same rules for inequalities hold as for real matrices. These structures also occur in other fields of mathematics [5], [6], [7]. They allow many theoretical considerations with floating-point matrices. The proposed implementation, furthermore, leads to a higher accuracy of floating-point matrix operations and allows a much simpler error analysis (Theorem 2.5).

Theorem 2.3 is the main result for the implementation. It reduces the structure of floating-point matrices to special properties of the rounding function and to a special definition of the operations. In chapter 1 these properties are derived as necessary conditions for an algebraic and order homomorphism between the real matrices and the floating-point matrices.

The last chapter gives the algorithms for the implementation of floating-point matrix operations for all roundings of the set $\{\nabla, \triangle, \square_\mu, \mu=0(1)b\}$ (for definition see chapter 1) using a special accumulator (Fig. 3 in chapter 3). It is an essential result that the implementation of all operations can be separated into several independent steps which means that an exchange of the rounding does not influence any other part of the algorithm.

Formalisierung und Implementierung von Gleitkommamatrixverknüpfungen. Die Arbeit zeigt, daß Gleitkommamatrixverknüpfungen in einer Weise implementiert werden können, welche eine vernünftige mathematische Struktur wie auch sinnvolle Verträglichkeitsbedingungen zwischen der Struktur der reellen Matrizen und derjenigen der Gleitkommamatrizen erlaubt. Es stellt sich beispielsweise heraus, daß alle Regeln für den Minusoperator bei reellen Matrizen erhalten werden können und daß für alle mit Null vergleichbaren Elemente bezüglich $\leqq$ und $\geqq$ dieselben Regeln für das Rechnen mit Ungleichungen gelten wie für reelle Matrizen. Die sich ergebenden Strukturen treten auch in anderen Gebieten der Mathematik auf [5], [6], [7]. Sie ermöglichen viele theoretische Untersuchungen mit Gleitkommamatrizen. Die vorgeschlagene Implementierung hat ferner eine höhere Genauigkeit der Verknüpfungen für Gleitkommamatrizen zur Folge und erlaubt eine wesentlich einfachere Fehleranalysis (Satz 2.5).

Der Satz 2.3 stellt das Hauptergebnis für die Implementierung dar. Er führt die Struktur der Gleitkommamatrizen zurück auf spezielle Eigenschaften der Rundungsfunktion und eine spezielle Definition der Verknüpfungen. Im ersten Abschnitt werden diese Eigenschaften als notwendige Bedingungen für einen algebraischen und Ordnungshomomorphismus zwischen den reellen Matrizen und den Gleitkommamatrizen hergeleitet.

* Dedicated to J. Weissinger on the occasion of his 62-nd birthday.

Der letzte Abschnitt behandelt die Algorithmen für die Implementierung der Verknüpfungen für Gleitkommamatrizen für alle Rundungsfunktionen der Menge $\{\triangledown, \triangle, \square_\mu, \mu=0(1)b\}$ (bezüglich der Definition siehe den ersten Abschnitt). Dabei wird ein spezieller Akkumulator verwendet (Abb. 3 im dritten Abschnitt). Ein wesentliches Ergebnis besteht darin, daß die Implementierung für alle Operationen in verschiedene, voneinander unabhängige Schritte zerlegt werden kann. Dies bedeutet beispielsweise, daß ein Austausch der Rundungsfunktion keinen anderen Teil des Gesamtalgorithmus beeinflußt.

1. Mathematical Preliminaries

To be clear let us first define a few well known order properties.

Definition 1.1: A set M with a relation $\leqq$ defined for its elements is called an "ordered set" $\{M, \leqq\}$ if it has the following properties:

$(O1)$ $\quad \bigwedge_{x\in M} x\leqq x$ (reflexive)

$(O2)$ $\quad \bigwedge_{x,y,z\in M} (x\leqq y \wedge y\leqq z \Rightarrow x\leqq z)$ (transitive)

$(O3)$ $\quad \bigwedge_{x,y\in M} (x\leqq y \wedge y\leqq x \Rightarrow x=y)$ (antisymmetric)

An ordered set is called "totally ordered" or "linearly ordered" if

$(O4)$ $\quad \bigwedge_{x,y\in M} (x\leqq y \vee y\leqq x)$ (linearly ordered)

holds.

In an ordered set the relation $<$ is defined as usual by

$$\bigwedge_{x,y\in M} (x<y :\Leftrightarrow x\leqq y \wedge x\neq y).$$

The real numbers $\mathbb{R}$ with respect to the addition $+$ the multiplication $\cdot$ and the order relation $\leqq$ have the following properties:

$(F1)$ $\{\mathbb{R}, +, \cdot\}$ is a field.

$(F2)$ $\{\mathbb{R}, \leqq\}$ is a linearly ordered set.

$(F3)$ The following compatibility properties hold between the algebraic and the order structure in $\mathbb{R}$:

(a) $\bigwedge_{x,y,z\in\mathbb{R}} (x<y \Rightarrow x+z<y+z)$,

(b) $\bigwedge_{x,y,z\in\mathbb{R}} (x<y \wedge z>o \Rightarrow x\cdot z<y\cdot z)$.

It is well known that every real number x can uniquely be represented by a b-adic expansion. Let us now give one of the usual definitions of a floating-point system:

Definition 1.2: A real number $x\in\mathbb{R}$ is called a "normalized floating-point number" if it fulfils the following conditions

$$\begin{gathered} x = m \cdot b^e \\ m := * o, d_1 d_2 \dots d_n, \; * \in \{+, -\} \\ 1 \leqq d_1 \leqq b-1, \; o \leqq d_i \leqq b-1 \text{ for all } i = 2\,(1)\,n, \\ e\,1 \leqq e \leqq e\,2, \; e\,1 \leqq o, \; e2 \geqq 1, \text{ where } e, e\,1, e\,2 \text{ are integers.} \end{gathered} \tag{1}$$

b is called the "base", m the "mantissa" or fraction part and e the "exponent" of the representation (1). The union of all normalized floating-point numbers with a special representation of the number zero which generally consists of a mantissa zero and the exponent $e\,1$:

$$+o.oo \dots o \cdot b^{e1} := o$$

is called a "floating-point system" $T \subseteq \mathbb{R}$.

A floating-point system therefore is characterized by four constants

$$T = T(b, n, e\,1, e\,2).$$

It consists of a finite number of elements and has the property

(S) $\quad o, 1 \in T \wedge \bigwedge_{x \in T} -x \in T$

which, among others, means that T is symmetric to zero.

Let now $M_r\,\mathbb{R}$ denote the set of real $r \times r$ matrices, O the zero matrix and E the unit matrix. Then $\{M_r\,\mathbb{R}, +, \cdot, \leqq\}$ with the usual addition and multiplication for matrices and the order relation defined by

$$\bigwedge_{X = (x_{ij}),\; Y = (y_{ij}) \in M_r\,\mathbb{R}} (X \leqq Y \Leftrightarrow x_{ij} \leqq y_{ij} \text{ for all } i, j = 1\,(1)\,r)$$

is an ordered ring with unit element E, i.e.

(M 1) $\quad \{M_r\,\mathbb{R}, +, \cdot\}$ is a ring with unit element E.

(M 2) $\quad \{M_r\,\mathbb{R}, \leqq\}$ is an ordered set.

(M 3) $\quad$ The following compatibility properties hold between the algebraic and the order structure in $M_r\,\mathbb{R}$:

(a) $\quad \bigwedge_{X, Y, Z \in M_r\,\mathbb{R}} (X < Y \Rightarrow X + Z < Y + Z).$

(b) $\quad \bigwedge_{X, Y, Z \in M_r\,\mathbb{R}} (X < Y \wedge Z > O \Rightarrow X \cdot Z < Y \cdot Z).$

Let now $M_r\,T$ denote the set of all $r \times r$ matrices with components out of a given floating-point system $T = T(b, n, e\,1, e\,2)$. Then also $M_r\,T$ consists of a finite number of elements and it also has the property

(S) $\quad O, E \in M_r\,T \wedge \bigwedge_{X \in M_r\,T} -X \in M_r\,T.$

This means that also $M_r\,T$ is symmetric to the zero matrix O.

Definition 1.3: Let M be a set and $S \subseteq M$. A mapping $\square : M \to S$ is called a "rounding" if it has the property

(*R*1) $\bigwedge_{x\in S} \square x = x.$

If $\{M, \leqq\}$ is a ordered set a rounding is called "monotone" if

(*R*2) $\bigwedge_{x,y\in M} (x \leqq y \Rightarrow \square x \leqq \square y).$ (monotone)

A rounding is called "directed" if

(*R*3) $\bigwedge_{x\in M} \square x \leqq x$ (downwardly directed)

$\vee \bigwedge_{x\in M} x \leqq \square x.$ (upwardly directed)

If in M a minusoperator is defined a rounding is called "antisymmetric" if

(*R*4) $\bigwedge_{x\in M} \square(-x) = -\square x.$ (antisymmetric)

Let now T again denote a given floating-point system. Then according to [7] we shall use special notations for the following special roundings from $\mathbb{R}$ into T:

$\bigwedge_{x\in\mathbb{R}} \nabla x \leqq x$ monotone downwardly directed rounding

$\bigwedge_{x\in\mathbb{R}} x \leqq \triangle x$ monotone upwardly directed rounding

$\bigwedge_{x\geqq o} \square_b x \leqq x \wedge \bigwedge_{x<o} \square_b x = -\square_b(-x)$ monotone rounding toward zero

$\bigwedge_{x\geqq o} x \leqq \square_o x \wedge \bigwedge_{x<o} \square_0 x = -\square_o(-x)$ monotone rounding away from zero

Further let

$$S_\mu(x) := \nabla x + \frac{(\triangle x - \nabla x)}{b} \cdot \mu, \ \mu = 1\,(1)\,b-1.$$

Then we define roundings $\square_\mu : \mathbb{R} \to T$, $\mu = 1\,(1)\,b-1$ by

$$\bigwedge_{x\in[o,\, b^{e1-1})} \square_\mu x = o$$

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

$$\bigwedge_{x<o} \square_\mu x = -\square_\mu(-x),$$

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

If b is an even number then $\square_{b/2} : \mathbb{R} \to T$ denotes the rounding to the nearest number of T:

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

Further special roundings $\square_*: \mathbb{R} \to T$ with the property

$$\bigwedge_{x \in \mathbb{R}} \left(|x| \in [b^{e1-1}, B] \Rightarrow \square_* x = x(1-\varepsilon) \text{ with } |\varepsilon| \leqq \varepsilon^* \right) \tag{2}$$

will be of interest where ε^* is a constant independent of x.

Theorem 1.1: *Let $\square : \mathbb{R} \to T$ be a rounding. If we define a mapping $\square : M_r \mathbb{R} \to M_r T$ by*

$$\bigwedge_{X=(x_{ij}) \in M_r \mathbb{R}} \square X := (\square x_{ij}) \tag{3}$$

then also $\square : M_r \mathbb{R} \to M_r T$ is a rounding. If further the rounding $\square : \mathbb{R} \to T$ is monotone, directed, antisymmetric respectively then also the rounding $\square : M_r \mathbb{R} \to M_r T$ is monotone, directed, antisymmetric respectively.

Above we have defined special notations for the special roundings of the set $\{\nabla, \Delta, \square_\mu, \mu = o(1)b, \square_*\}$ where each element was a mapping of $\mathbb{R}$ into T. Using (3) we can now generalize the same symbols and define mappings $\{\nabla, \Delta, \square_\mu, \mu = o(1)b, \square_*\}$ of $M_r \mathbb{R}$ into $M_r T$. Then it is easy to see that the rounding $\square_* : M_r \mathbb{R} \to M_r T$ has the property

$$\bigwedge_{X=(x_{ij}) \in M_r \mathbb{R}} \left(\bigwedge_{i,j} |x_{ij}| \in [b^{e1-1}, B] \Rightarrow \square_* X = \left(x_{ij}(1-\varepsilon_{ij})\right) \Rightarrow \right. \\ \left. \Rightarrow |X - \square_* X| \leqq \varepsilon^* |X| \right) \tag{4}$$

where $|\varepsilon_{ij}| < \varepsilon^*$ and ε^* is defined by (2) and the absolute value is defined componentwise.

The central question of the following chapters will be how the operations for elements of $M_r T$ are to be defined. In order to find an answer for this question we are now going to derive a few plausibility reasons by similar considerations than in [7].

A structure $\{M, \bar{M}\}$ is defined by a set M of elements and a set $\bar{M}$ of rules (axioms) given for the elements of M.

Definition 1.4: Let $\{M, \bar{M}\}$ and $\{T, \bar{T}\}$ 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 $\square : M \to T$ is called a "homomorphism" if it is an algebraic hòmomorphism, i. e. if

$$\bigwedge_{X, Y \in M} (\square X) \boxed{*} (\square Y) = \square (X * Y) \tag{5}$$

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

$$\bigwedge_{X, Y \in M} (X \leqq Y \Rightarrow \square X \leqq \square Y). \tag{6}$$

If X, Y are elements of $M_r T \subseteq M_r \mathbb{R}$ and $*$ is one of the operations $+, -, \cdot$ for elements of $M_r \mathbb{R}$ then in general $X * Y$ is not again an element of $M_r T$. Therefore the operations $* \in \{+, -, \cdot\}$ for real matrices on a computer only can be approximated. From algebra it is known that the homomorph image of a ring is again a ring. On the other hand we know already that the operations $+$ or $\cdot$ even for floating-point numbers are no longer associative. We come, therefore, immediately to the conclusion that it is impossible to realize an homomorphism between the two ordered algebraic structures $\{M_r \mathbb{R}, \overline{M_r \mathbb{R}}\}$ and $\{M_r T, \overline{M_r T}\}$. We shall see, however, that it is possible to implement a few necessary conditions for an homomorphism. Doing this we go as far to an homomorphism as possible. We are now going to derive these necessary conditions.

We already mentioned that the set $M_r T$ has the property

$$(S) \qquad O, E \in T \wedge \bigwedge_{X \in M_r T} -X \in M_r T.$$

Further a rounding $\square : M_r \mathbb{R} \to M_r T$ was defined by

$$(R\,1) \qquad \bigwedge_{X \in M_r T} \square X = X.$$

Let us now for a moment assume that the rounding $\square : M_r \mathbb{R} \to M_r T$ would define an homomorphism between the ordered algebraic structures $\{M_r \mathbb{R}, \overline{M_r \mathbb{R}}\}$ and $\{M_r T, \overline{M_r T}\}$. If we then for $M = M_r \mathbb{R}$ restrict (5) to elements of $M_r T$ we get immediately because of $(R\,1)$

$$(R) \qquad \bigwedge_{X, Y \in M_r T} X \boxed{*} Y = \square (X * Y).$$

We shall use this formula later to define the operation $\boxed{*}$, $* \in \{+, -, \cdot\}$, in $M_r T$ by the corresponding operations $* \in \{+, -, \cdot\}$ in $M_r \mathbb{R}$ and the rounding function $\square : M_r \mathbb{R} \to M_r T$.

From (6) we get for $M = M_r \mathbb{R}$ immediately that the rounding has to be a monotone function

$$(R\,2) \qquad \bigwedge_{X = (x_{ij}) \in M_r \mathbb{R}} (X \leqq Y \Rightarrow \square X \leqq \square Y).$$

If we further replace $X = -E$ in (5) we get in the case of multiplication:

$$\bigwedge_{Y \in M_r \mathbb{R}} \square (-Y) = \square (-E) \boxdot \square Y \underset{(S), (R\,1)}{=} (-E) \boxdot \square Y = \square (-\square Y) \underset{(R)}{=} \ \underset{(S), (R\,1)}{=} -\square Y, \text{ i.e.}$$

$$(R\,4) \qquad \bigwedge_{X \in M_r \mathbb{R}} \square (-X) = -\square X.$$

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

The conditions $(R\,1)$, $(R\,2)$ and $(R\,4)$ do not define the rounding function uniquely. For instance we have already seen that the special roundings $\square_\mu : M_r \mathbb{R} \to M_r T$, $\mu = o\,(1)\,b$, which we have defined above all have these properties. However, as we shall see in the next chapters, these properties together with formula (R) for the operations of floating-point matrices do already define the structure $\{M_r T, \overline{M_r T}\}$ of the floating-point matrices.

It still remains the question whether an arithmetic for floating-point matrices which fulfills all our assumptions $(R\,1)$, $(R\,2)$, $(R\,4)$ and (R) can be implemented on computers by fast algorithms. We shall answer this question positively within the last chapter for all roundings of the set $\{\nabla, \triangle, \square_\mu, \mu = o\,(1)\,b\}$.

2. The Structure $\{M_r T, \overline{M_r T}\}$ of Floating-Point Matrices

Let us first summarize our main results. We have seen that the floating-point matrices $M_r T$ have the property

$$(S) \qquad O, E \in M_r T \wedge \bigwedge_{X \in M_r T} -X \in M_r T.$$

A rounding $\square : M_r \mathbb{R} \to M_r T$ was defined by

$$(R1) \qquad \bigwedge_{X \in M_r T} \square X = X.$$

As necessary conditions for an homomorphism between $M_r \mathbb{R}$ and $M_r T$ we have derived the further properties

$$(R2) \qquad \bigwedge_{X \in M_r \mathbb{R}} (X \leqq Y \Rightarrow \square X \leqq \square Y) \qquad \text{monotone}$$

$$(R4) \qquad \bigwedge_{X, Y \in M_r \mathbb{R}} \square(-X) = -\square X \qquad \text{antisymmetric}$$

and

$$(R) \qquad \bigwedge_{X, Y \in M_r T} X \mathbin{\boxed{*}} Y := \square(X * Y) \quad \text{for all} \quad * \in \{+, -, \cdot\}.$$

We are now going to define the special structure of an ordered ringoid and shall later prove that this under the assumptions (S), $(R1)$, $(R2)$, $(R4)$ and (R) is the structure of the floating-point matrices.

Definition 2.1: A set S in which an addition, a multiplication and an order relation are defined is called an "ordered ringoid" if

$$(D1) \qquad \bigwedge_{X, Y \in S} X + Y = Y + X$$

$$(D2) \qquad \bigvee_{O \in S} \bigwedge_{X \in S} X + O = X$$

$$(D3) \qquad \bigvee_{E \in S \setminus \{O\}} \bigwedge_{X \in S} X \cdot E = E \cdot X = X$$

$$(D4) \qquad \bigwedge_{X \in S} X \cdot O = O \cdot X = O$$

$(D5)$ There exists an element $\Psi \in S \setminus \{E\}$ such that:

(a) $\Psi \cdot \Psi = E$

(b) $\bigwedge_{X, Y \in S} \Psi \cdot (X \cdot Y) = (\Psi \cdot X) \cdot Y = X \cdot (\Psi \cdot Y)$

(c) $\bigwedge_{X, Y \in S} \Psi \cdot (X + Y) = \Psi \cdot X + \Psi \cdot Y.$

$(D6)$ Ψ is unique.

$\{S, \leqq\}$ is an ordered set, i.e. the order properties $(O1)$, $(O2)$, $(O3)$ hold as well as the following compatibility properties between the algebraic and the order structure:

(*OD*1) $\bigwedge_{X,Y,Z\in S} (X \leqq Y \Rightarrow X+Z \leqq Y+Z)$

(*OD*2) $\bigwedge_{X,Y\in S} (X \leqq Y \Rightarrow \Psi \cdot Y \leqq \Psi \cdot X)$

(*OD*3) $\bigwedge_{X,Y,Z\in S} (O \leqq X \leqq Y \wedge Z \geqq O \Rightarrow X \cdot Z \leqq Y \cdot Z \wedge Z \cdot X \leqq Z \cdot Y)$.

We are now using the uniqueness of Ψ for the following

Definition 2.2: Let $\{S, +, \cdot, \leqq\}$ be an ordered ringoid. Then we define a minus-operator and a subtraction by

$$\bigwedge_{X\in S} -X := \Psi \cdot X \tag{1}$$

$$\bigwedge_{X,Y\in S} X - Y := X + (-Y). \tag{2}$$

Simple consequences:

(1) $\underset{X=E}{\Rightarrow} \Psi = -E$

($D5a$) $\Rightarrow (-E)\cdot(-E) = E$

($D5b$) $\Rightarrow -(X \cdot Y) = (-X) \cdot Y = X \cdot (-Y)$

($D5c$) $\Rightarrow -(X+Y) = (-X) + (-Y)$

($OD2$) $\Rightarrow (X \leqq Y \Rightarrow -Y \leqq -X)$.

Theorem 2.1: *In an ordered ringoid* $\{S, +, \cdot, \leqq\}$ *the following properties hold:*

(*a*) $E \neq O$, $-E \neq O$, $-E \neq E$.

Further the same rules for the minus-operator hold as in $\mathbb{R}$, *for instance:*

(*b*) $\bigwedge_{X\in S} O - X = -X$

(*c*) $\bigwedge_{X\in S} -X = (-E)\cdot X = X \cdot (-E)$

(*d*) $\bigwedge_{X\in S} -(-X) = X$

(*e*) $\bigwedge_{X,Y\in S} -(X-Y) = -X + Y = Y - X$

(*f*) $\bigwedge_{X,Y\in S} (-X)\cdot(-Y) = X \cdot Y$

(*g*) *O respectively E is the only neutral element of the addition resp. multiplication.*

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

Furthermore in an ordered ringoid for all elements which are comparable with O with respect to the relations $\leqq$ *and* $\geqq$ *the same rules for inequalities hold as in* $\mathbb{R}$, *for instance*

(*i*) $\bigwedge_{X,Y,U,V\in S} (X \leqq Y \wedge U \leqq V \Rightarrow X + U \leqq Y + V)$

(j) $\bigwedge_{X,Y,U,V\in S} (O\leqq X\leqq Y \wedge O\leqq U\leqq V \Rightarrow O\leqq XU\leqq YV \wedge O\leqq UX\leqq VY)$

(k) $\bigwedge_{X,Y,U,V\in S} (X\leqq Y\leqq O \wedge U\leqq V\leqq O \Rightarrow XU\geqq YV\geqq O \wedge UX\geqq VY\geqq O)$

(l) $\bigwedge_{X,Y,U,V\in S} (X\leqq Y\leqq O \wedge O\leqq U\leqq V \Rightarrow XV\leqq YU\leqq O \wedge VX\leqq UY\leqq O)$.

Proof:

(a): follows immediately from (D3), (D4) and (D5).

(b): $O-X=O+(-X)=-X$.

(c): from ($D5b$) we get with $Y=E$ immediately $-X=(-E)\cdot X=X\cdot(-E)$.

(d): $-(-X)=(-E)((-E)X) \underset{(D5b)}{=} ((-E)(-E))X \underset{(D5a)}{=} X.$

(e): $-(X-Y)=(-E)(X+(-Y)) \underset{(D5c)}{=} (-E)X+(-E)((-E)Y) \underset{(d)}{=} -X+Y=Y-X.$

(f): $(-X)(-Y)=((-E)X)((-E)Y) \underset{(D5b)}{=} (-E)(((-E)X)Y) \underset{(D5b)}{=} ((-E)(-E))XY \underset{(D5a)}{=} XY.$

(g): we assume that E' would be another neutral element of the multiplication, then $E=E\cdot E'=E'\cdot E=E'$.
The proof for the addition follows analogously.

(h): $(D4)\Rightarrow X-O=X+(-O)=X+O=X$, i.e. O is right neutral element of the subtraction. Let us assume O' would be another right neutral element of the subtraction. Then because of definition 2.2:

$$\bigwedge_{X\in S} X-O'=X+(-O')=X \underset{(g)}{\Rightarrow} -O'=O \underset{(D4)}{\Rightarrow} -(-O')=O \underset{(d)}{\Rightarrow} O'=O.$$

(i): $(OD1)\Rightarrow X+U\leqq Y+U\leqq Y+V$

(j): $(OD3)\Rightarrow O\leqq XU\leqq YU\leqq YV$

(k): $(OD2)\Rightarrow O\leqq -Y\leqq -X \wedge O\leqq -V\leqq -U \underset{(j),(D5b)}{\Rightarrow} XU\geqq YV\geqq O$

(l): $(OD2)\Rightarrow O\leqq -Y\leqq -X \wedge O\leqq U\leqq V \underset{(j)}{\Rightarrow} O\leqq -(YU)\leqq -(XV) \underset{(OD2)}{\Rightarrow} XV\leqq YU\leqq 0.$ ■

Remark: In general there do not exist inverse elements of the addition in an ordered ringoid. But nevertheless the subtraction is no independent operation. It can be defined by the multiplication and the addition.

Theorem 2.2: *Let* $\{S, +, \cdot, \leqq\}$ *be an ordered ringoid with the special elements* $\{-e, o, e\}$ *and let us by* $M_r S$ *denote the set of all* $r\times r$ *matrices with components out of* S. *In* $M_r S$ *the equality, addition, multiplication and order relation are defined by the usual formulas over the components. Then also* $\{M_r S, +, \cdot, \leqq\}$ *is an ordered ringoid with the special elements* $\{\Psi, O, E\}$ *with*

$$\Psi=-E:=\begin{pmatrix} -e & o & \dots & o \\ o & -e & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & & -e \end{pmatrix}, O:=\begin{pmatrix} o & o & \dots & o \\ o & o & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & \dots & o \end{pmatrix}, E:=\begin{pmatrix} e & o & \dots & o \\ o & e & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & \dots & e \end{pmatrix}.$$

Proof: (see [5] and [9]) $(D1)$, $(D2)$, $(D3)$ and $(D4)$ are immediately clear. Further, the matrix $\Psi = -E$ fulfills $(D5a)$. It is also easy to see that for all $X, Y \in M_r S$ the properties

$(D5b)$: $\Psi \cdot (X \cdot Y) = (\Psi \cdot X) \cdot Y = X \cdot (\Psi \cdot Y)$ and

$(D5c)$: $\Psi \cdot (X + Y) = (\Psi \cdot X) + (\Psi \cdot Y)$

hold. In order to prove $(D6)$ we have to show that $\Psi = -E$ is the only matrix of $M_r S$ which satisfies $(D5)$.

Let $\Psi = (\zeta_{ij}) \neq E$ be any element of $M_r S$ which satisfies $(D5)$. Then we get by $(D5b)$ with $Y = E$ for all $X \in M_r S$: $\Psi \cdot X = X \cdot \Psi$, i.e. the element commutes with all $X \in M_r S$. For X we now choose special matrices $X^k := (x_{ij}^k)$, $k = 1\,(1)\,r$, with

$$x_{ij}^k := \begin{cases} e & \text{for } i, j = k \\ o & \text{else.} \end{cases}$$

Then from $\Psi \cdot X^k = X^k \cdot \Psi$ follows immediately $\zeta_{ik} = \zeta_{ki} = o$ for $i \neq k$, $i, k = 1\,(1)\,r$. That means that Ψ is a diagonal matrix. Now we choose a matrix $X = (x_{ij})$ with $x_{ij} = e$ for all $i, j = 1\,(1)\,r$. Then from $\Psi \cdot X = X \cdot \Psi$ follows $\zeta_{ii} = \zeta_{jj}$ for all $i, j = 1\,(1)\,r$. This means that all diagonal elements of Ψ are equal. We denote this element by ζ. Let now x, y be arbitrary elements of S and X, Y the matrices

$$X = \begin{pmatrix} x & o & \dots & o \\ o & x & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & \dots & x \end{pmatrix}, \quad Y = \begin{pmatrix} y & o & \dots & o \\ o & y & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & \dots & y \end{pmatrix}.$$

If we then apply $(D5a, b, c)$ of $M_r S$ for these matrices Ψ, X, Y we get

a) $\zeta \cdot \zeta = e$

b) $\bigwedge\limits_{x, y \in S} \zeta \cdot (x \cdot y) = (\zeta \cdot x) \cdot y = x \cdot (\zeta \cdot y)$

c) $\bigwedge\limits_{x, y \in S} \zeta \cdot (x + y) = (\zeta \cdot x) + (\zeta \cdot y)$

This means that the diagonal element ζ of Ψ fulfills $(D5a, b, c)$ in S. Since $\Psi \neq E$ it is $\zeta \neq e$. By $(D6)$ in S therefore $\zeta = -e$ and $(D6)$ in $M_r S$ is proved.

We still have to prove that in $M_r S$ also the order- and the compatibility properties hold. Since $(O1)$, $(O2)$, $(O3)$, $(OD1)$ and $(OD2)$ are easily to see we just prove

$(OD3)$: $O \leqq X \leqq Y \wedge Z \geqq O \Rightarrow o \leqq x_{ij} \leqq y_{ij} \wedge z_{ij} \geqq o$ for all $i, j = 1\,(1)\,r \Rightarrow$

$$\underset{(OD3)_S}{\Rightarrow} o \leqq x_{i\nu}\, z_{\nu j} \leqq y_{i\nu}\, z_{\nu j} \text{ for all } i, j, \nu = 1\,(1)\,r \underset{\text{Theorem 2.1 } (i)}{\Rightarrow} X \cdot Z \leqq Y \cdot Z. \quad \blacksquare$$

Remark: Theorem 2.2 allows many applications. Let for instance $\mathbb{R}$ be the real number field, T a given floating-point system and S another floating-point system let us say of double length. Then $\mathbb{R} \supseteq S \supseteq T$ and it is proved in [7] Theorem 5.1 that, if the arithmetic in S and T is properly implemented, $\{\mathbb{R}, +, \cdot, \leqq\}$, $\{S, +, \cdot, \leqq\}$ and $\{T, +, \cdot, \leqq\}$ are all ordered ringoids. It follows by Theorem 2.2, therefore, that also $\{M_r\,\mathbb{R}, +, \cdot, \leqq\}$, $\{M_r\,S, +, \cdot, \leqq\}$ and $\{M_r\,T, +, \cdot, \leqq\}$ are ordered ringoids (Fig. 1).

$$\begin{tikzcd}
\mathbb{R} \arrow[r, "\text{[7] Theorem 5.1}"] \arrow[d, "\text{Theorem 2.2}"] & S \arrow[r, "\text{[7] Theorem 5.1}"] \arrow[d, "\text{Theorem 2.2}"] & T \arrow[d, "\text{Theorem 2.2}"] \\
M_r\,\mathbb{R} & M_r\,S & M_r\,T
\end{tikzcd}$$

Fig. 1. Definition of the operations in M_rS and M_rT by Theorem 2.2

This remark describes exactly the way how matrix-operations on computers are usually defined. The matrix-operations in $M_r\,T$ for instance are defined by the floating-point operations in T and in general there are no obvious compatibility properties valid between the matrix operations in $M_r\,S$ or $M_r\,T$ and $M_r\,\mathbb{R}$.

We shall in the last chapter propose another implementation of floating-point matrix operations by fast algorithms which leads to such compatibility properties as well as to a much higher accuracy of floating-point matrix computations. This implementation is based on

Theorem 2.3: *Let $\{S, +, \cdot, \leqq\}$ be an ordered ringoid with the special elements $\{-e, o, e\}$ and $\{M_r\,S, +, \cdot, \leqq\}$ an ordered ringoid in the set of matrices over S with the special elements $\{-E, O, E\}$. Let further T be a subset of S with the property (S), $\square: S \to T$ a rounding with the properties $(R1)$, $(R2)$, $(R4)$ and $\{T, \boxplus, \boxdot, \leqq\}$ again an ordered ringoid with the same special elements $\{-e, o, e\}$ where the operations in T are defined by the rounding $\square$ and formula (R).*

Then the set $M_r\,T$ of matrices over T has also the property (S) and the mapping $\square: M_r\,S \to M_r\,T$ which is defined by

$$\bigwedge_{X=(x_{ij})\in M_r\,S} \square X := (\square\, x_{ij})$$

is a rounding with the properties $(R1)$, $(R2)$, $(R4)$. If in $M_r\,T$ operations $\boxed{*}$, $* \in \{+, \cdot\}$ are defined by

$$(R) \qquad \bigwedge_{X,\,Y\in M_r\,T} X \boxed{*} Y := \square\,(X * Y) \quad \textit{for all} * \in \{+, \cdot\}$$

then $\{M_r\,T, \boxplus, \boxdot, \leqq\}$ also is an ordered ringoid with the special elements $\{-E, O, E\}$. Further between the structure in $M_r\,S$ and that in $M_r\,T$ the following compatibility properties hold:

$$(RG1) \qquad \bigwedge_{X,\,Y\in M_r\,T} (X * Y \in M_r\,T \Rightarrow X \boxed{*} Y = X * Y) \quad \textit{for all} * \in \{+, -, \cdot\}$$

$$(RG2) \qquad \bigwedge_{X,\,Y,\,U,\,V\in M_r\,T} (X * Y \leqq U * V \Rightarrow X \boxed{*} Y \leqq U \boxed{*} V) \quad \textit{for all} * \in \{+, -, \cdot\}.$$

Proof: The properties $(D1)$, $(D2)$, $(D3)$, $(D4)$ and $(D5)$ of $M_r\,T$ can easily be verified. As an example we prove the property

$$(D5c):\quad (-E) \boxdot X \underset{(R)}{=} \square\,((-E)\cdot X) = \square\,(-X) \underset{(R4)}{=} -\square\, X \underset{(R1)}{=} -X. \qquad (3)$$

$$(-E)\boxdot(X\boxplus Y)\underset{(R)}{=}\square((-E)\cdot\square(X+Y))\underset{(R4)}{=}\square(\square(-(X+Y)))\underset{(R1)}{=}$$
$$=\square(-(X+Y))\underset{(D5c)_{M_r S}}{=}\square((-X)+(-Y))\underset{(R)}{=}(-X)\boxplus(-Y)\underset{(3)}{=}$$
$$=((-E)\boxdot X)\boxplus((-E)\boxdot Y).$$

The proof of $(D6)$ follows similar steps than the proof of $(D6)$ in Theorem 2.2.

$(D6)$: We have to show that $-E$ is the only element in $M_r T$ which fulfills $(D5a, b, c)$. Let $\Psi=(\zeta_{ij})\neq E$ be any element of $M_r T$ which satisfies $(D5)$. Then we get from $(D5b)$ with $Y=E$ for all $X\in M_r T$: $\Psi\boxdot X=X\boxdot\Psi$, i.e. the element Ψ commutes with all $X\in M_r T$. For X we now choose the special matrices $X^k:=(x_{ij}^k)$, $k=1\,(1)\,r$, with

$$x_{ij}^k:=\begin{cases} e & \text{for } i,j=k \\ o & \text{else.}\end{cases}$$

Then from $\Psi\boxdot X^k=X^k\boxdot\Psi$ follows immediately $\zeta_{ik}=\zeta_{ki}=o$ for $i\neq k$, $i,k=1\,(1)\,r$. This means that Ψ is a diagonal matrix. Now we choose a matrix $X=(x_{ij})$ with $x_{ij}=e$ for all $i,j=1\,(1)\,r$. Then from $\Psi\boxdot X=X\boxdot\Psi$ follows $\zeta_{ii}=\zeta_{jj}$ for all $i,j=1\,(1)\,r$. This means that all diagonal elements of Ψ are equal to an element ζ.

Let now x, y be arbitrary elements of T and X, Y the matrices

$$X=\begin{pmatrix} x & o & \dots & o \\ o & x & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & \dots & x\end{pmatrix},\quad Y=\begin{pmatrix} y & o & \dots & o \\ o & y & \dots & o \\ \dots & \dots & \dots & \dots \\ o & o & \dots & y\end{pmatrix}.$$

If we now apply $(D5a, b, c)$ of $M_r T$ for these matrices Ψ, X, Y we get

a) $\zeta\boxdot\zeta=e$

b) $\bigwedge_{x,y\in T}\zeta\boxdot(x\boxdot y)=(\zeta\boxdot x)\boxdot y=x\boxdot(\zeta\boxdot y)$

c) $\bigwedge_{x,y\in T}\zeta\boxdot(x\boxplus y)=(\zeta\boxdot x)\boxplus(\zeta\boxdot y)$,

where $\boxplus$ and $\boxdot$ denote the addition and multiplication in the ringoid $\{T,\boxplus,\boxdot,\leqq\}$. This means that the diagonal element ζ of Ψ fulfills $(D5a, b, c)$ in T. Since $\Psi\neq E$ it is $\zeta\neq e$. By $(D6)$ in T therefore $\zeta=-e$ and $(D6)$ in $M_r T$ is proved.

We still have to prove that in $M_r T$ also the order- and compatibility properties hold. As a subset of $M_r S$ $M_r T$ of course is again an ordered set, $(OD1)$ is a simple consequence of $(OD1)$ in $M_r S$, (R) and $(R2)$.

$(OD2)$: $$\bigwedge_{X\in M_r T}(-E)\boxdot X\underset{(R)}{=}\square((-E)\cdot X)=\square(-X)\underset{(R4)}{=}-\square X\underset{(R1)}{=}-X. \qquad (4)$$

$$X\leqq Y\underset{(OD2)_{M_r S}}{\Rightarrow}-Y\leqq -X\underset{(4)}{\Rightarrow}(-E)\boxdot Y\leqq(-E)\boxdot X\Rightarrow\boxminus Y\leqq\boxminus X.$$

$$(OD3):\quad O \leqq X \leqq Y \wedge Z \geqq O \underset{(OD3)_{M_r S}}{\Rightarrow} X \cdot Z \leqq Y \cdot Z \wedge Z \cdot X \leqq Z \cdot Y \underset{(R2),(R)}{\Rightarrow}$$

$$\Rightarrow X \boxdot Z \leqq Y \boxdot Z \wedge Z \boxdot X \leqq Z \boxdot Y.$$

The compatibility properties $(RG1)$ and $(RG2)$ are simple consequences of $(R1)$, $(R2)$ and (R). $(RG1)$ and $(RG2)$ for the subtraction can be proved by the definitions (1) and (2), (4) and the corresponding properties for the addition. ■

Remark: Theorem 2.3 allows many applications. Let again $\mathbb{R}$ be the real number field, T a given floating-point system and S another floating-point system for instance the system with double length in the mantissas. Then $\mathbb{R} \supseteq S \supseteq T$. If the mappings from $\mathbb{R}$ into S respectively from S into T are monotone and antisymmetric roundings and the operations in S respectively T are defined by formula (R) then it is proved by [7] Theorem 5.1 that $\{\mathbb{R}, +, \cdot, \leqq\}$, $\{S, +, \cdot, \leqq\}$ and $\{T, +, \cdot, \leqq\}$ are all ordered ringoids. Theorem 2.2 proves that also $\{M_r\, \mathbb{R}, +, \cdot, \leqq\}$ is an ordered ringoid. If we now define operations in $M_r\, S$ by formula (R) Theorem 2.3 proves that also $\{M_r\, S, +, \cdot, \leqq\}$ becomes an ordered ringoid with compatibility properties $(RG1)$ and $(RG2)$. Now we can repeat this step and define operations in $M_r\, T$ by the operations in $M_r\, S$ and formula (R). Then by Theorem 2.3 also $\{M_r\, T, +, \cdot, \leqq\}$ becomes an ordered ringoid and again the compatibility properties $(RG1)$ and $(RG2)$ hold (Fig. 2).

$$\begin{tikzcd}
\mathbb{R} \arrow[r, "\text{[7] Theorem 5.1}"] \arrow[d, "\text{Theorem 2.2}"] & S \arrow[r, "\text{[7] Theorem 5.1}"] & T \\
M_r\, \mathbb{R} \arrow[r, "\text{Theorem 2.3}"] & M_r\, S \arrow[r, "\text{Theorem 2.3}"] & M_r\, T
\end{tikzcd}$$

Fig. 2. Definition of the operations in $M_r S$ and $M_r T$ by Theorem 2.3

The ringoids in $M_r\, S$ and $M_r\, T$ which we mentioned in this remark differ by the definition of the operations essentially from those which we have discussed in the remark after Theorem 2.2. The following considerations show that floating-point matrix computations using the operations of Theorem 2.3 lead to much higher accuracy and allow a much simpler error analysis than those using the operations of Theorem 2.2.

Without giving the proof we mention the following well known

Theorem 2.4: *Let $T = T(b, n, e1, e2)$ be a floating-point system and $\square : \mathbb{R} \to T$ a monotone rounding and let $\delta(\square\, x) := x - \square\, x$ denote the absolute rounding error and $\varepsilon := \delta(\square\, x)/x$ the relative rounding error. Then*

$$\bigwedge_{x \in \mathbb{R}} \left(b^{e1-1} \leqq |x| \leqq B \Rightarrow \square\, x = x(1-\varepsilon) \text{ with } |\varepsilon| < \varepsilon^* \Rightarrow |x - \square\, x| \leqq \varepsilon^* \cdot |x|\right) \tag{5}$$

where

$$\varepsilon^* := \begin{cases} \frac{1}{2} b^{1-n} & \text{for } \square = \bar{\square} \\ b^{1-n} & \text{for } \square \neq \bar{\square} \end{cases} \tag{6}$$

is independent of x and $\bar{\Box} : \mathbb{R} \to T$ denotes the rounding to the nearest floating-point number.

If we apply this theorem to floating-point operations we get immediately the following

Corollary: *Let $T = T(b, n, e1, e2)$ be a floating-point system, $\Box : \mathbb{R} \to T$ a monotone rounding and*

$$(R) \qquad \bigwedge_{x, y \in T} x \boxed{*} y := \Box(x * y) \;\; \textit{for all} \;\; * \in \{+, -, \cdot, /\}.$$

Let further $\delta(x \boxed{*} y) := x * y - (x \boxed{*} y)$ be the absolute rounding error and $\varepsilon := \delta(x \boxed{*} y)/(x * y)$ the relative rounding error. Then we get for all $* \in \{+, -, \cdot, /\}$

$$\bigwedge_{x, y \in T} \Big(b^{e1-1} \leqq |x * y| \leqq B \Rightarrow x \boxed{*} y = (x * y)(1 - \varepsilon) \textit{ with } |\varepsilon| < \varepsilon^* \Rightarrow \\ \Rightarrow |x * y - x \boxed{*} y| \leqq \varepsilon^* \cdot |x * y|\Big) \qquad (7)$$

where ε^* is defined by (6).

Theorem 2.4 and the Corollary are 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 correctly implemented. Error estimations for floating-point matrix computations have to take into account that the matrix multiplication is defined by scalar products. If the floating-point matrix operations are defined in the sense of Theorem 2.2 by the basic floating-point operations, therefore, we get relatively complicated and inaccurate formulas. If we define the floating-point matrix operations, however, by Theorem 2.3 we get by corresponding considerations which led to the Corollary above immediately the

Theorem 2.5: *Let $\mathbb{R}$ be the real number field, $\{M_r \mathbb{R}, +, \cdot, \leqq\}$ the real matrices, $T = T(b, n, e1, e2)$ a floating-point system and $\Box : \mathbb{R} \to T$ a monotone and antisymmetric rounding and let in T operations be defined by formula (R). Then by [7] Theorem 5.1 also $\{T, \boxplus, \boxdot, \leqq\}$ is an ordered ringoid. If we now define a rounding $\Box : M_r \mathbb{R} \to M_r T$ by*

$$\bigwedge_{X = (x_{ij}) \in M_r \mathbb{R}} \Box X := (\Box x_{ij})$$

then

$$\bigwedge_{X = (x_{ij}) \in M_r \mathbb{R}} \Big(\bigwedge_{i,j} |x_{ij}| \in [b^{e1-1}; B] \Rightarrow \Box X = \big(x_{ij}(1 - \varepsilon_{ij})\big) \Rightarrow |X - \Box X| \leqq \varepsilon^* \cdot |X|\Big) \qquad (8)$$

where $B := o \cdot (b-1)(b-1) \dots (b-1) \cdot b^{e2}$, $|\varepsilon_{ij}| < \varepsilon^*$ and ε^* is defined by (6) and the absolute value is defined componentwise.

If in $M_r T$ operations $\boxed{*}$, $* \in \{+, \cdot\}$ are defined by

$$(R) \qquad \bigwedge_{X, Y \in M_r T} X \boxed{*} Y := \Box(X * Y)$$

then with $Z := (z_{ij}) := X * Y$ for all $* \in \{+, -, \cdot\}$ the following formula holds:

$$\bigwedge_{X,Y\in M_rT} \Big(\bigwedge_{i,j} |z_{ij}| \in [b^{e1-1}; B] \Rightarrow X \boxed{*}\, Y = \big(z_{ij}(1-\varepsilon_{ij})\big) \Rightarrow \tag{9}$$
$$\Rightarrow |X * Y - X \boxed{*}\, Y| \leqq \varepsilon^* \cdot |X * Y|\Big).$$

Proof: The proof follows immediately from the definition of the matrix operations and formula (4) in chapter 1. ■

The results of (5), (7), (8), (9) respectively can simply be transformed into

$$\begin{aligned} x-\varepsilon^*|x| &\leqq \square x \leqq x+\varepsilon^*|x| \\ x*y-\varepsilon^*|x*y| &\leqq x \boxed{*}\, y \leqq x*y+\varepsilon^*|x*y| \\ X-\varepsilon^*|X| &\leqq \square X \leqq X+\varepsilon^*|X| \\ X*Y-\varepsilon^*|X*Y| &\leqq X \boxed{*}\, Y \leqq X*Y+\varepsilon^*|X*Y| \end{aligned}$$

Remark: Theorem 2.5 remains valid if $\mathbb{R}$ respectively $M_r\mathbb{R}$ are replaced by an ordered ringoid in a floating-point system $\{S, +, \cdot, \leqq\}$ respectively $\{M_rS, +, \cdot, \leqq\}$. Then, for instance, it covers also the mapping from a floating-point matrix system M_rS with a mantissa of n digits into a system M_rT with a $n_1 < n$ digit mantissa.

3. Implementation of Floating-Point Matrix Operations

We are now going to discuss the question whether and how the structures which we have derived in the last chapter can be implemented on computers by fast algorithms. Theorem 2.3 is the basic theorem for the implementation. It reduces the structure of floating-point matrices to a special definition of the operations and special properties of the rounding function. In chapter 1 we have derived these properties as necessary conditions for an homomorphism.

By Theorem 2.3 formula (R) the operations $\boxed{*}$, $* \in \{+, \cdot\}$, in M_rT have to be defined by

$$(R) \qquad \bigwedge_{X,Y\in M_rT} X \boxed{*}\, Y := \square(X * Y) \text{ for all } * \in \{+, \cdot\}.$$

If $X = (x_{ij})$ and $Y = (y_{ij})$ then in case of the addition we get

$$X \boxplus Y := \square(X + Y) = (x_{ij} \boxplus y_{ij}).$$

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

$$X \boxdot Y := \square(X \cdot Y) := \square\left(\sum_{\nu=1}^{r} x_{i\nu}\, y_{\nu j}\right) \tag{1}$$

where in

$$\sum_{\nu=1}^{r} x_{i\nu}\, y_{\nu j} \tag{2}$$

the multiplications and additions denote the real multiplication and addition. In order to generate (1), (2) seems to be necessary. But (2) even on computers with a so called accumulator of double length is only very seldom exactly representable. It will turn out, however, that in such cases it is sufficient to replace (2) by an appropriate and representable value

$$\widetilde{\sum_{\nu=1}^{r} x_{i\nu}\, y_{\nu j}} \tag{3}$$

with the property

$$\square\,(X\cdot Y)=\square\left(\sum_{\nu=1}^{r} x_{i\nu}\, y_{\nu j}\right)=\square\,(X\tilde{\cdot}Y)=\square\left(\widetilde{\sum_{\nu=1}^{r} x_{i\nu}\, y_{\nu j}}\right). \tag{4}$$

Then (4) can be used to define $X \boxdot Y$. By the following algorithms we shall especially prove this assertion.This proof should be understood as an existence-proof. Although it is given by a realizable and fast algorithm even faster algorithms might possibly exist.

In order to realize (4) we first have to calculate the products $x_{i\nu}\cdot y_{\nu j}$. If x_{ij} and y_{ij} are floating-point numbers of n digits in the mantissa $x_{i\nu}\cdot y_{\nu j}$ can exactly be generated within an accumulator of $L=2\,n$ digits. Let us assume that this is done. Then (4) can be generated if we show that the sum

$$z:=\square\left(\sum_{i=1}^{r} x_i\right)=\square\left(\widetilde{\sum_{i=1}^{r} x_i}\right) \tag{5}$$

can be implemented on the computer where the x_i, $i=1\,(1)\,r$, denote $L=2\,n$ digit floating-point numbers and z is an n digit floating-point number. It should be mentioned that the algorithm which we shall discuss below also could be used to produce a floating-point number z defined by (5) of $n, n+1, \ldots, L=2\,n$ correct digits just by rounding the intermediate result $\widetilde{\sum_{i=1}^{r} x_i}$ to other lengths.

The following algorithms give the implementation of formula (5). The whole algorithm can in principle be separated into the following nine steps:

1. Decomposition of x_i, $i=1\,(1)\,r$, i.e. separation of x_i into exponent part e_i and mantissa m_i.
2. Elimination of zeros among the x_i, $i=1\,(1)\,r$.
3. Execution of the exponent order $e_1 \geqq e_2 \geqq \ldots \geqq e_r$.
4. Execution of a strong exponent order $e_1 > e_2 > \ldots > e_r$.
5. Addition from left to right.
6. Addition from right to left.
7. Normalization.
8. Rounding.
9. Composition, i.e. combination of the resulting exponent part and mantissa to a floating-point number.

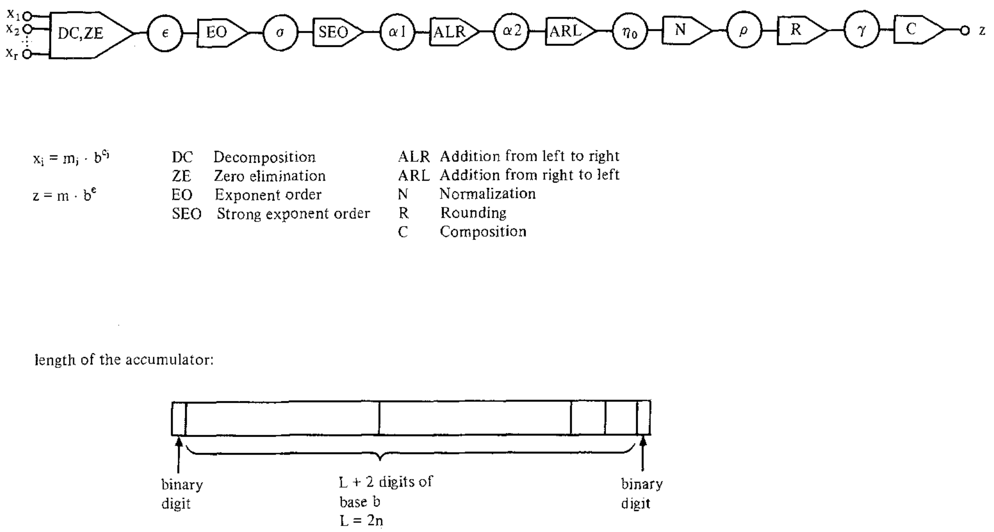

Fig. 3. Flow diagram of the summation $z = \square\left(\widetilde{\sum_{i=1}^{r} x_i}\right) = \square\left(\sum_{i=1}^{r} x_i\right)$, and length of the accumulator

Fig. 3 gives a graphical diagram of these nine steps. Between these steps labels are used to denote the cuts of the explicit flow diagrams which we shall discuss below. The whole algorithm uses an accumulator of one digit which can be a binary digit in front of the point and $L+2$ digits of base b plus one further binary digit after the point. If n denotes the number of digits of the floating-point mantissa then $L=2\,n$.

The algorithms have especially to take care that the formulas ($R1$), ($R2$), ($R4$) and (R) are strictly realized. This means that they really have to be valid for all $X, Y \in M_r\, T$ respectively $M_r\, \mathbb{R}$ and not only for some of them or mostly. Under these requirements a further reduction of the length of the accumulator should be difficult respectively impossible.

The following algorithms give the implementation of formula (5) for all roundings of the set $\{\nabla, \triangle, \square_\mu, \mu = o\,(1)\,b\}$. It is an essential result that the whole implementation can be separated into nine *independent* steps as indicated in Fig. 3 and its context. This means that the intermediate result $\widetilde{\sum_{i=1}^{r}} x_i$ can be chosen independently of the rounding function such that

$$\bigwedge_{\square \in \{\nabla, \triangle, \square_\mu, \mu = o\,(1)\,b\}} \ \bigwedge_{x_i \in T} \square\left(\sum_{i=1}^{r} x_i\right) = \square\left(\widetilde{\sum_{i=1}^{r} x_i}\right).$$

In the flow diagram of Fig. 3, therefore, instead of the rounding R any one of the roundings $\square \in \{\nabla, \triangle, \square_\mu, \ \mu = o\,(1)\,b\}$ can be substituted and we get the correct result which is defined by this rounding and (5) without changing any other part of the algorithm. Explicit algorithms for the steps 8 (rounding) and 9 (composition) are not given in this paper. With respect to these algorithms we refer to [7].

In the following flow diagrams, as usual, rectangles denote statements; circles, lables; and figures with six edges, conditions. A further special and obvious symbol is used to denote the "for statement". The input numbers x_i, $i = 1\,(1)\,r$, are denoted by $x_i = m_i \cdot b^e$ the output by $z = m \cdot b^e$. The k-th digit of m_i respectively m is denoted by $m_i\,[k]$ respectively $m\,[k]$.

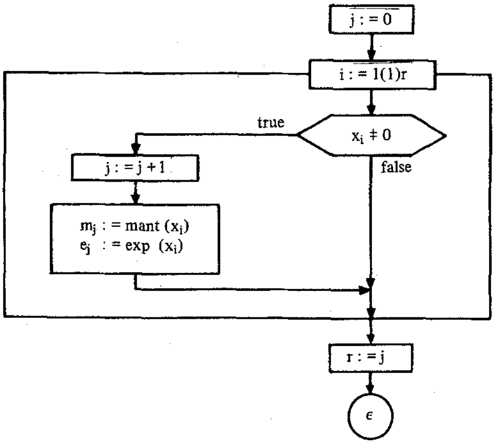

Fig. 4. Decomposition and zero elimination

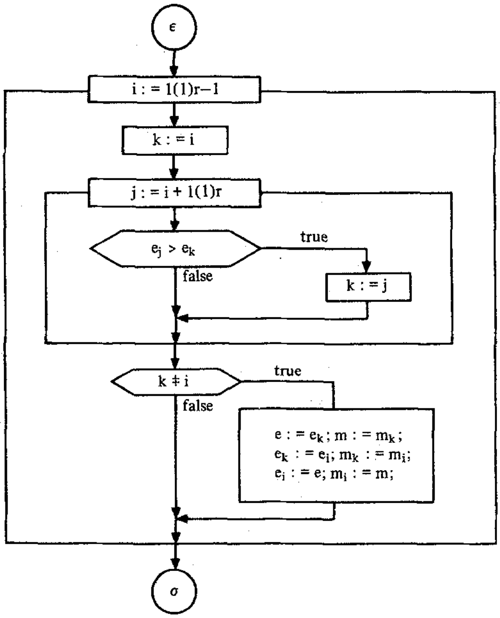
ε
i : = 1(1)r–1
k : = i
j : = i + 1(1)r
$e_j > e_k$
true
false
k : = j
$k \neq i$
true
false
e : = e_k; m : = m_k;
e_k : = e_i; m_k : = m_i;
e_i : = e; m_i : = m;
σ

Fig. 5. Execution of exponent order $e_1 \geqq e_2 \geqq \ldots \geqq e_r$

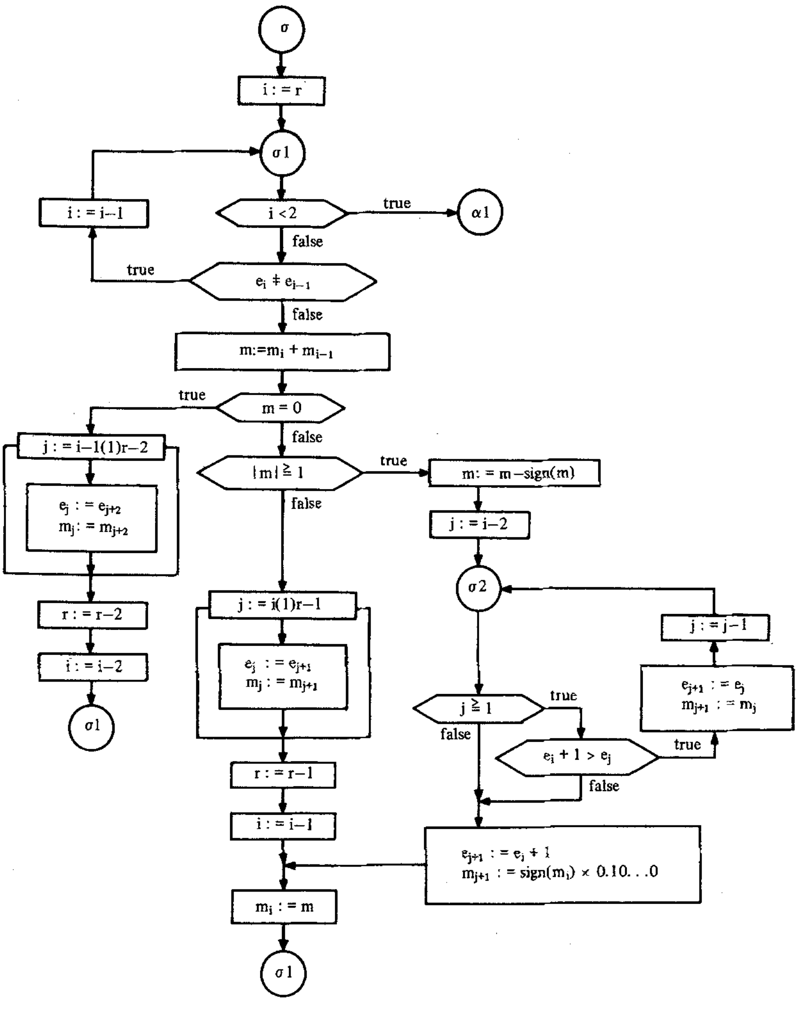

Fig. 6. Execution of a strong exponent order $e_1 > e_2 > \ldots > e_r$

The algorithm works similarly than the "algorithm for the execution of the addition with a short accumulator" as given in [7] where for our purposes n is replaced by L. Since we have to add not only two but r floating-point numbers of length L we first order the x_i by their exponents. Figs. 4, 5 and 6 give the algorithms for the decomposition, zero elimination, the execution of an exponent order and a strong exponent order. After these steps the x_i, $i=1\,(1)\,r$, are ordered into a strong decreasing chain of their exponents

$$e_1 > e_2 > \ldots > e_r. \qquad (6)$$

The mantissas m_i of the x_i, $i=1\,(1)\,r$, have no carry. But they can have leading zeros.

r elements x_i, $i=1\,(1)\,r$, with the property (6) could be added in a natural way without accumulation of rounding errors by addition from right to left.

The whole sum

$$\sum_{i=1}^{r} x_i, \tag{7}$$

however, can be suffering from catastrophic cancellation. Since in general the magnitude of the result z is not known at the beginning it is difficult to estimate the number of digits over which this addition would have to be carried out. We begin, therefore, the whole addition with an addition from left to right as described in Fig. 7. It delivers the magnitude of the result z. If no further addition from left to right by the algorithm of Fig. 7 is possible the mantissa m has no leading zeros (but possibly a carry) and $L+2$ digits of base b on the right hand side of the point. If $k-1$ elements of the x_i are already added we have

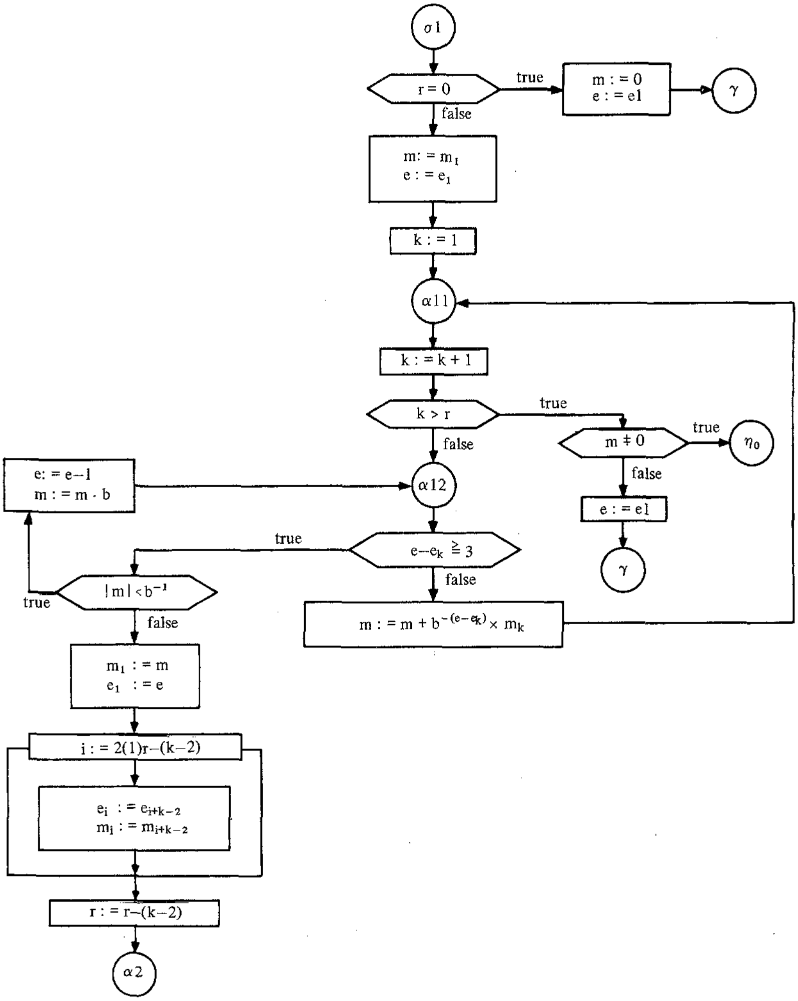

Fig. 7. Addition from left to right

$$e - e_k \geqq 3 \tag{8}$$

and

$$\left| m \cdot b^e + \sum_{i=k}^{r} m_i \cdot b^{e_i} \right| \geqq b^{e-1} - \sum_{i=k}^{r} b^{e_i} \underset{(6)}{\geqq} b^{e-1} - b^{e_k} \sum_{i=o}^{r-k} b^{-i} >$$

$$> b^{e-1} - b^{e_k} \sum_{i=o}^{\infty} b^{-i} \underset{(8)}{\geqq} b^{e-1} - b^{e-3} \frac{1}{1-b^{-1}} \geqq b^{e-1} - b^{e-2} \geqq b^{e-2}.$$

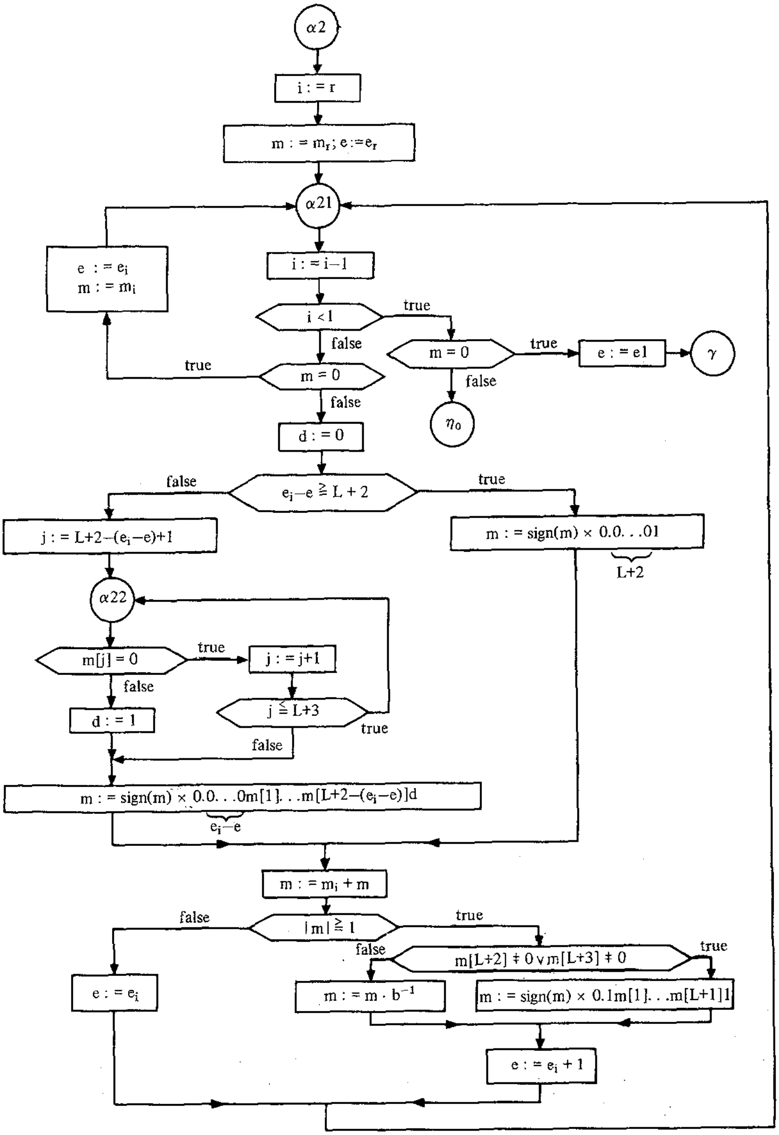

Fig. 8. Addition from right to left

I.e. in the whole sum at most a cancellation of one further digit can occur and the rest can be added from right to left as described in Fig. 8. We use for this purpose again the accumulator which we have described in Fig. 3. In order to allow a simple description we have assumed in the algorithm of Fig. 8 that the $(L+3)$-rd digit is a full digit of base b. It could, however, be represented by a single binary digit.

The algorithm in Fig. 8 is following the strategy that after every addition the intermediate or final result m is correct within the possible carry digit and the first $L+2$ digits on the right hand side of the point. The $(L+3)$-rd digit on the right hand side of the point carries the information which is necessary to get the correct results in all cases of the roundings $\nabla, \triangle, \square_o$ and $\square_b$.

Since after the addition from left to right m in general has already $L+2$ digits of base b the case $e_i - e \leqq 2$ can not be treated in a separate and simpler branch as in the case of an addition of two elements of length L.

Fig. 9 finally gives the algorithm for the normalization.

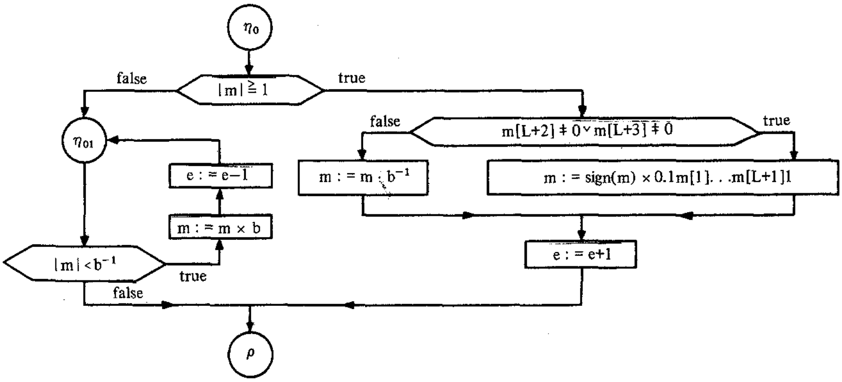

Fig. 9. Normalization

References

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

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

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

[4] 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).

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

[6] Kulisch, U.: Grundlagen des Numerischen Rechnens. Niederschrift einer vierstündigen Vorlesung, gehalten im WS 1970/71 an der Universität Karlsruhe, S. 1—250.

[7] Kulisch, U.: Implementation and Formalization of Floating-Point Arithmetics, IBM T. J. Watson-Research Center, Report Nr. RC 4608, Nov. 1973, p. 1—50, published in part in Computing under the title: Formalization and Implementation of Floating-Point Arithmetics.

[8] Kulisch, U.: Über die Arithmetik von Rechenanlagen, Universität Karlsruhe, Aug. 1974, S. 1—74, to appear in „Überblicke Mathematik 1975". Mannheim-Wien-Zürich: Bibliographisches Institut.

[9] Ullrich, Ch.: Rundungsinvariante Strukturen mit äußeren Verknüpfungen. Dr.-Dissertation, Universität Karlsruhe, 1972, p. 1—67.

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

[11] Yohé, J. M.: Roundings in Floating-Point Arithmetic. IEEE Transactions on Computers, Vol. C-22, No. 6, June 1973.

Prof. Dr. U. Kulisch
Dipl.-Math. G. Bohlender
Institut für Angewandte Mathematik
Universität Karlsruhe
Kaiserstraße 12
D-7500 Karlsruhe
Federal Republic of Germany