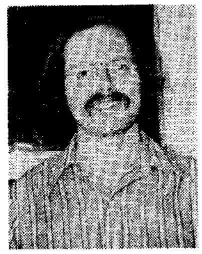
tions," in *Conf. Rec. Ass. Comput. Mach. Theory of Computing*, Marina del Rey, Calif., May 1969.
[9] R. Brent, "On the addition of binary numbers," *IEEE Trans. Comput.* (Short Notes), vol. C-19, pp. 758–759, Aug. 1970.
[10] I. Marques, private communication.
[11] A. Avižienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 389–400, Sept. 1961.
[12] —, "On the problem of computational time and complexity of arithmetic functions," in *Conf. Rec. Ass. Comput. Mach. Theory of Computing*, Marina del Rey, Calif., May 1969.
[13] Z. Kohavi, *Switching and Finite Automata Theory.* New York: McGraw-Hill, 1970.



Philip M. Spira was born in Worcester, Mass., on June 30, 1941. He received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, in 1963 and the M.S. and Ph.D. degrees from Stanford University, Stanford, Calif., in 1965 and 1968, respectively, all in electrical engineering.

He is currently teaching computer science at the University of California, Berkeley. His research is mainly concerned with complexity of computation and algorithms.

The Status of Investigations into Computer Hardware Design Based on the Use of Continued Fractions

JAMES E. ROBERTSON AND KISHOR S. TRIVEDI

Abstract—**The purpose of this paper is to demonstrate that representations of numbers other than positional notation may lead to practical hardware realizations for digital calculation of classes of algorithms. This paper describes current research in the use of continued fractions. Although practicality has not been demonstrated, theoretical results are promising.**

Index Terms—**Computer arithmetic, continued fractions, continued products, hardware, quadratic equation, radix, representation of numbers, Riccati equation, selection rules.**

I. History and Motivation

This paper is essentially a report on research in progress. The fundamental observation is that, currently, virtually all digital hardware calculations are based on the use of positional notation; equivalently, on weighted sums of series. Other representations of numbers exist; the concern here will be with continued products and continued fractions.

The use of positional notation has been limited to addition, subtraction, multiplication, division, and, to a lesser extent, square and higher roots. It has been shown [1] that use of continued products extends the list of implementable algorithms to the logarithm, the exponential, the trigonometric and inverse trigonometric functions, as well as multiply, divide, and square root. Both time of execution and cost of hardware are reasonable with current technology; in comparison with a conventional arithmetic unit, factors of 2 to 3 (depending on the function) for both time and cost are typical. A small read-only memory fast enough to match accumulator rates is also needed. The investigation of the use of continued products was originally limited to the binary case. Higher radix techniques appear promising, and are being investigated [2]. Otherwise, emphasis will be given here to investigations into the use of continued fractions. Results to date are theoretically promising, but not yet practical in the sense of hardware implementation.

There appear to be three fundamental requirements for a proposed representation of numbers to be useful for implementation in hardware. These are the following.

Requirement 1: Conversion to conventional series form (positional notation) must be both possible and simple. Implicit here is the requirement that the set of possible results spans continuously (in the limit of infinite precision) some permissible range of values. For floating-point arithmetic, it seems sufficient to require that the ratio of the upper limit to the lower limit of the range be at least two.

Requirement 2: The set of algorithms should include algorithms that are easily soluble for the representation of numbers employed. Compatibility among algorithms, in the sense of hardware sharing, is also a desirable goal.

Requirement 3: Since most algorithms, other than multiplication, appear to require trial and error procedures in the absence of redundancy, it must be possible to devise techniques such that the selection of each of the successive coefficients is practical (cf., quotient-digit selection in division).

It should be pointed out that the use of the coefficients of a representation is ephemeral, since conversion to positional

Manuscript received August 11, 1972; revised November 21, 1972. This work was supported in part by NSF Grant GJ813, in part by the Applied Mathematics Department of Brookhaven National Laboratory, and in part by the Department of Computer Science, University of Illinois, Urbana, Ill.

The authors are with the Department of Computer Science, University of Illinois, Urbana, Ill. 61801.

notation occurs in parallel with the successive steps of the algorithm. For example, for a continued product

$$\prod_{i=1}^{k} (1 + 2^{-i}\epsilon_i) = (1 + 2^{-k}\epsilon_k) \prod_{i=1}^{k-1} (1 + 2^{-i}\epsilon_i).$$

At any one step the $(k-1)$st continued product has been determined, the coefficient ϵ_k is determined by the selection rules appropriate to the algorithm, and the kth value of the continued product (in positional notation) is calculated by adding the $(k-1)$st value to a shifted version of itself. The entire set of coefficients ϵ_i $(i = 1, 2, \cdots, m)$ is never simultaneously provided.

It is difficult to generalize about the procedures necessary to determine whether or not a proposed representation satisfies the requirements previously discussed. For continued products and continued fractions, determination of the set of coefficient values and the associated conversion procedure has been relatively simple. The identification of suitable algorithms appears to be by far the most difficult requirement to satisfy. In retrospect, for continued products, the observation that the logarithm of a continued product is the sum of the logarithms of the individual terms leads to the identification of the logarithm and its inverse, the exponential, as suitable algorithms. Similarly, the properties of the complex exponential indicate that the trigonometric functions and their inverses are identifiable as algorithms for continued product representations. No such general observation is as yet apparent to the authors for continued fractions. Formulating selection rules appears to be very much a function of the individual algorithm.

II. Examples: Division Algorithms

For illustrative purposes, algorithms for division are developed for positional notation, for continued products, and for continued fractions. In each case, the initial assumption is that

$$\frac{N}{D} - X \to 0$$

or some variant thereof, where N is the dividend, D is the divisor, and X is the quotient. The same procedures are then used for developing the algorithms, except for the representation of X. For positional notation the division algorithm in common use is developed. For continued products, a new algorithm with many useful properties is found. The continued fraction algorithm is obviously an exercise in futility, since the conversion procedure requires of itself a division as its terminal step.

A. Positional Notation

We define the remainder at the ith step by

$$N - DX_i = Y_i$$

and also

$$N - DX_{i-1} = Y_{i-1}.$$

For positional notation

$$X_i = X_{i-1} + 2^{-i}x_i = \sum_{j=1}^{i} 2^{-j}x_j$$

$$Y_i = N - DX_i = N - DX_{i-1} - 2^{-i}Dx_i = Y_{i-1} - 2^{-i}Dx_i.$$

Since the allowed range of Y_i decreases by a factor of two at each step, it is convenient to define a shifted remainder r_i:

$$r_i = 2^i Y_i$$

and also

$$r_{i-1} = 2^{i-1} Y_{i-1}$$

$$r_i = 2^i Y_{i-1} - Dx_i = 2r_{i-1} - Dx_i. \tag{1}$$

Equation (1) is the familiar recursion for most binary division procedures in common use. The initial remainder $Y_0 = r_0 = N$, the dividend, and $X_0 = 0$.

B. Continued Products

As in the previous example, the remainder is

$$N - DX_i = Y_i$$

and

$$N - DX_{i-1} = Y_{i-1}.$$

For a continued product

$$X_i = X_{i-1}(1 + 2^{-i}x_i) = \prod_{j=1}^{i} (1 + 2^{-j}x_j)$$

$$\begin{aligned} Y_i = N - DX_i &= N - DX_{i-1} - 2^{-i}DX_{i-1}x_i \\ &= Y_{i-1} - 2^{-i}(N - Y_{i-1})x_i \\ &= Y_{i-1}(1 + 2^{-i}x_i) - 2^{-i}Nx_i. \end{aligned} \tag{2}$$

It is again convenient to define a shifted remainder r_i.

$$r_i = 2^i Y_i$$

and

$$\begin{aligned} r_{i-1} &= 2^{i-1} Y_{i-1} \\ r_i &= 2^i Y_{i-1}(1 + 2^{-i}x_i) - Nx_i \\ &= 2r_{i-1}(1 + 2^{-i}x_i) - Nx_i. \end{aligned} \tag{3}$$

The conversion procedure is

$$X_i = X_{i-1} + 2^{-i}x_i X_{i-1}$$

with $X_0 = 1$ and $x_i = \overline{1}, 0, 1$, or $x_i = 0, 1$. The initial remainder is $Y_0 = r_0 = N - D$.

An alternative that simplifies the selection procedure is to let $N = 1$ in (3), and compensate by letting $X_0 = N$ as the initial condition for the conversion procedure.

C. Continued Fractions

For continued fractions, let $X_i = P_i/Q_i$, and define the remainder Y_i as

$$Y_i = NQ_i - DP_i.$$

The conversion procedure is given by the recursions

$$P_i = q_i P_{i-1} + p_i P_{i-2}, \qquad P_0 = 0 \quad P_1 = p_1$$

$$Q_i = q_i Q_{i-1} + p_i Q_{i-2}, \qquad Q_0 = 1 \quad Q_1 = q_1 \tag{4}$$

which must be followed by a terminal division ($i = m$) as indicated by $X_m = P_m/Q_m$. Otherwise, the conversion consists of additions and shifts if p_i and q_i are simple binary coefficients; e.g., $\frac{1}{4}$, $\frac{1}{2}$, 1, and 2.

We note that

$$Y_{i-2} = NQ_{i-2} - DP_{i-2}$$
$$Y_{i-1} = NQ_{i-1} - DP_{i-1}$$

therefore,

$$Y_i = N(q_i Q_{i-1} + p_i Q_{i-2}) - D(q_i P_{i-1} + p_i P_{i-2})$$
$$Y_i = q_i Y_{i-1} + p_i Y_{i-2}. \tag{5}$$

Equation (5) is derived here for illustrative purposes only. Due to the obvious impracticality of the process, neither the rate of convergence (i.e., decrease in range of Y_i) nor the selection procedure (i.e., method of choosing q_i and p_i) have been studied.

III. The First Quadratic

Consider the finite continued fraction with k partial numerators p_i and k partial denominators q_i ($i = 1, 2, \cdots, k$), whose value is P_k/Q_k, i.e.,

$$\frac{P_k}{Q_k} = \cfrac{p_1}{q_1 + \cfrac{p_2}{q_2 + \cfrac{p_3}{q_3 + \ddots + \cfrac{p_k}{q_k}}}}.$$

P_k and Q_k are determined from the recursions

$$P_i = q_i P_{i-1} + p_i P_{i-2}, \qquad i = 2, 3, \cdots, k$$
$$Q_i = q_i Q_{i-1} + p_i Q_{i-2}, \qquad P_0 = 0 \quad P_1 = p_1 \quad Q_0 = 1 \quad Q_1 = q_1.$$

It is clear that P_k and Q_k can be separately and simultaneously determined in two binary arithmetic units in $k - 1$ addition times if the p_i and q_i are chosen to be simple in the binary sense. It is convenient to make the choice $p_i = 1$ for all i; it can be shown (Section VI) that other values of p_i are admissible. The digit set for q_i is initially assumed to be two-valued, and after some investigation it was found that choice of the digit set $q_i = \{\frac{1}{2}, 1\}$ yields continued fractions whose values P_k/Q_k are continuous in the limit over the interval as defined by the following equation:

$$\frac{1}{2} \leqslant \lim_{k \to \infty} \frac{P_k}{Q_k} \leqslant 1.$$

These properties indicate that a suitable continued fraction representation exists, such that conversion to conventional binary can be achieved by repetitive use of two binary adders in parallel, followed by a division to determine the quotient P_k/Q_k.

Determination of an algorithm and the appropriate corresponding computational procedure is much more difficult. The particular problem chosen for investigation was the solution of the limited class of quadratics

$$x^2 + b_k x - c_k = (x - u)(x + v) = 0 \tag{6}$$

such that $\frac{1}{2} \leqslant u \leqslant 1$. The problem, specifically, is, given b_k and c_k, find u (and hence $v = b_k + u$). This problem was selected because of the following property of infinite periodic continued fractions, of period k. If the value of the first $k - 1$ terms is P_{k-1}/Q_{k-1} and the value of the first k terms is P_k/Q_k, then the quadratic coefficients b_k and c_k are $b_k = (Q_k - P_{k-1})/Q_{k-1}$ and $c_k = P_k/Q_{k-1}$. The value of the infinite periodic continued fraction is then u, the positive root of the quadratic. The problem is then resolved specifically to the following one. Given $(Q_k - P_{k-1})/Q_{k-1}$ and P_k/Q_{k-1} (note that k is unknown), find the sequence of partial denominators q_i ($i = 1, 2, \cdots, k$).

The first three approaches, which for brevity cannot be described here, were abandoned as impossible or impractical for mechanization. The fourth approach led to a relatively simple computational procedure.

The fourth approach was based on the observation that the value u of the infinite periodic continued fraction of period k with $p_i = 1$, $q_i = \{\frac{1}{2}, 1\}$, ($i = 1, 2, \cdots, k$) is also the value u of the infinite periodic continued fraction of period one with each $p_i = c_k$ and each $q_i = b_k$ ($i = 1, 2, \cdots, \infty$). That is,

$$u = \frac{c_k}{b_k + u}.$$

This approach may therefore be considered as a method of conversion from one form of an infinite continued fraction to that form which is easily converted to the conventional binary representation.

The fourth approach, in successive steps, generates partial quotients q_1, q_2, etc., by increasing the periodicity of periodic continued fractions as follows:

$$u = \cfrac{c_k}{b_k + \cfrac{c_k}{b_k + \cfrac{c_k}{b_k + \ddots}}} = \cfrac{1}{q_1 + \cfrac{c_{k-1}}{b_{k-1} + \cfrac{1}{q_1 + \cfrac{c_{k-1}}{b_{k-1} + \ddots}}}}$$

$$= \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{c_{k-2}}{b_{k-2} + \ddots}}}$$

After a considerable amount of algebra, the recursion relations can be shown to be

$$b_{k-n} = q_n c_{k-n+1} - q_{n-1} c_{k-n+2} + b_{k-n+2}$$
$$c_{k-n} = -q_n b_{k-n} + q_n b_{k-n+1} + c_{k-n+2}. \tag{7}$$

For $n = 1$, the recursions require that $b_{k+1} = 0$, $c_{k+1} = 1$, and $q_0 = 0$. Although it was initially felt that the requirement

$p_i = 1$ was necessary, later investigation, described in Section VI, has shown that values of p_i other than 1 may be used.

IV. Extension of the Range and Domain of Quadratic Solutions

In the previous section, the generality of the solution of the quadratic of (6) is limited by the requirement that the root u is representable. For the choice $q_i \in \{\frac{1}{2}, 1\}$ and $p_i = 1$, the range of u is $\frac{1}{2} \leqslant u \leqslant 1$. Replacing x in (6) by $u_{\min} = \frac{1}{2}$ and $u_{\max} = 1$, the solutions are limited to the triangular wedge in the c_k, b_k plane

$$\frac{1}{2} b_k + \frac{1}{4} \leqslant c_k \leqslant b_k + 1. \quad (8)$$

It will be shown in Section V that selection procedures impose the further requirement $b_k \geqslant 0$ (9). The purpose of this section is to show that any point in the upper half of the c_k, b_k plane (i.e., $c_k > 0$) can be mapped onto a point in the region defined by conditions (8) and (9).

At this point, it is convenient to delineate four areas in the c_k, b_k plane and relate each area to properties of the root magnitudes u and v.

1) $c_k < -(b_k^2)/(4)$. Both roots are imaginary.

2) $-(b_k^2)/(4) \leqslant c_k < 0$. Both roots are real and of the same sign.

3) $c_k \geqslant 0$, $b_k < 0$ (second quadrant). The roots are real and of opposite sign, with $u > v$.

4) $c_k \geqslant 0$, $b_k \geqslant 0$ (first quadrant). The roots are real and of opposite sign, with $v \geqslant u$.

It is first shown that any point in the first quadrant of the c_k, b_k plane may be scaled to lie within a triangular wedge such that $\frac{1}{2} \leqslant u \leqslant 1$. Since $v = u + b_k$ and $c_k = uv$, it follows that $c_k = ub_k + u^2$, and the range $\frac{1}{2} \leqslant u \leqslant 1$ is equivalent to

$$\frac{1}{2} b_k + \frac{1}{4} \leqslant c_k \leqslant b_k + 1. \quad (9)$$

Multiplying (6) and (8) by 2^{2j} (j an integer) yields

$$(2^j x)^2 + (2^j b_k)(2^j x) - 2^{2j} c_k = 0 \quad (10)$$

$$2^{j-1}(2^j b_k) + 2^{2(j-1)} \leqslant 2^{2j} c_k \leqslant 2^j (2^j b_k) + 2^{2j}. \quad (11)$$

Let $2^j x = x'$, $2^j b_k = b_k'$, and $2^{2j} c_k = c_k'$. Then

$$(x')^2 + b_k' x' - c_k' = 0 \quad (12)$$

$$2^{j-1} b_k' + 2^{2(j-1)} \leqslant c_k' \leqslant 2^j b_k' + 2^{2j}. \quad (13)$$

Given c_k' and b_k', the scaling procedure is then as follows.

1) Determine the value of j, such that (13) is satisfied.

2) Multiply c_k' and b_k' by 2^{-2j} and 2^{-j}, respectively, to obtain c_k and b_k, which satisfy (8).

3) When the root u is determined, find the positive root u' of (12), by scaling u in accordance with $u' = 2^j u$.

Note that the scaling procedure reduces to that normally employed for square roots in floating-point computers when $b_k = 0$.

For any point c_k', b_k' in the first quadrant, an integer value of j can be found such that (13) is satisfied. It is therefore sufficient, for the first quadrant, to solve (6) subject to the constraints of (8), with $b_k \geqslant 0$.

For the second quadrant, with $b_k < 0$ it is sufficient to replace $b_k = v - u$ by $b_k'' = -b_k = u - v$. Equation (6) becomes

$$x^2 + b_k'' x - c_k = (x - v)(x + u) = 0. \quad (14)$$

Solution of (14) yields the magnitude v of the negative root. The value of u is then $u = b_k'' + v$.

The solution for the case of two imaginary roots has not been considered. Attempts to find a method of solution for two real roots of the same sign have thus far been unsuccessful. The preceding observations, however, indicate that a continued fraction solution of the quadratic can be found if $c_k > 0$; i.e., if the two roots are real and are of opposite sign.

V. Selection Procedures for the First Quadratic

This section develops a selection procedure for p_i and q_i of the algorithm of (7) for solving quadratics using continued fractions. We decide to have $p_i = 1$ for all i. Thus the problem reduces to the selection of q_i.

First, we must choose the set from which to pick q_i; we call this a digit set of q_i. We put five requirements on this digit set.

Requirement 1: All elements must be of the form 2^j where j is an integer.

Requirement 2: Let the range of numbers representable as infinite continued fractions using this digit set be $[a, b]$. We require that this range form a continuum between a and b.

Requirement 3: The range $[\frac{1}{2}, 1]$ should be a subset of the range $[a, b]$.

Requirement 4: The cardinality of the digit set should be as low as possible.

Requirement 5: It should be possible to develop a selection procedure for our algorithm with this digit set.

The set $\{1, 2\}$ does not satisfy Requirement 2. The set $\{1, \frac{1}{2}\}$ satisfies all requirements except Requirement 5. The reason for this is that, with this set, every number representable as an infinite continued fraction, has a unique representation; as will be seen later in this paper, that a certain amount of redundancy in representation is necessary to satisfy Requirement 5. The set $\{1, \frac{1}{2}, \frac{1}{4}\}$ satisfies all five requirements; so now we focus our attention on this digit set.

Requirement 1 is clearly satisfied. It is easily shown that the range $[a, b]$, is approximately $[0.39, 1.56]$ with this digit set. Thus Requirement 3 is satisfied. Requirement 4 is also satisfied. To show that Requirement 2 is satisfied we can proceed as follows.

First any number $f_1 \in [a, b]$ can be expanded as a continued fraction as follows. Let

$$f_1 = \frac{1}{q_1 + f_2}$$

and in general, let

$$f_i = \frac{1}{q_i + f_{i+1}}.$$

If $a \leqslant f_i < \frac{1}{2}$ then choose $q_i = 1$.
If $\frac{1}{2} \leqslant f_i < 1$ then choose $q_i = \frac{1}{2}$.
If $1 \leqslant f_i \leqslant b$ then choose $q_i = \frac{1}{4}$.

It can easily be verified that with $f_1 \in [a, b]$ and using the above rules, $f_i \in [a, b]$ for all i. Therefore, the above rules can be used for all $i \geqslant 1$. We call such a method of expansion a consistent method of expansion. By an expansion of f_1 to k

terms is meant the fraction $1/q_1 + 1/q_2 + \cdots + 1/q_k$. Next we use the following theorem which we state without proof.

Theorem 1: For a number $f_1 \in [a, b]$, if there is a consistent method of expansion of f_1 in the form of a continued fraction, then such an expansion converges to the value f_1 as the number of terms in the expansion increases, provided that the smallest element in the digit set is greater than 0 [5].

Thus every number in $[a, b]$ has an infinite continued fraction expansion with the digit set $\{1, \frac{1}{2}, \frac{1}{4}\}$ and hence Requirement 2 is satisfied.

We devote the rest of this section to show that Requirement 5 is satisfied.

We restrict the problem to $b_k \geq 0$.

Let $f_i = c_{k-i}/(b_{k-i} + u)$ be expanded to $f_i = 1/(q_{i+1} + f_{i+1})$. Given that $0.39 \leq f_i \leq 1.56$, we have to find $q_{i+1} \in \{1, \frac{1}{2}, \frac{1}{4}\}$ such that $0.39 \leq f_{i+1} \leq 1.56$. From these, we get the following.

1) For $0.39\,(b_{k-i} + u) \leq c_{k-i} \leq 0.72\,(b_{k-i} + u)$; choose $q_{i+1} = 1$.

2) For $0.485\,(b_{k-i} + u) \leq c_{k-i} \leq 1.124\,(b_{k-i} + u)$; choose $q_{i+1} = \frac{1}{2}$.

3) For $0.553\,(b_{k-i} + u) \leq c_{k-i} \leq 1.56\,(b_{k-i} + u)$; choose $q_{i+1} = \frac{1}{4}$.

The regions where two choices are allowed are as follows.

1) $0.485\,(b_{k-i} + u) \leq c_{k-i} \leq 0.72\,(b_{k-i} + u)$ then $q_{i+1} = \frac{1}{2}$ or 1.

2) $0.553\,(b_{k-i} + u) \leq c_{k-i} \leq 1.124\,(b_{k-i} + u)$ then $q_{i+1} = \frac{1}{4}$ or $\frac{1}{2}$.

Both these are triangular wedges in the (c_{k-i}, b_{k-i}) plane. We will call these the ($\frac{1}{2}$ and 1) and the ($\frac{1}{4}$ and $\frac{1}{2}$) overlap regions, respectively. Clearly these wedges vary with u. To get a selection line that decides between $q_{i+1} = \frac{1}{2}$ or 1 and which is u independent, (since u is unknown) we should first take the intersection of all ($\frac{1}{2}$ and 1) regions as u varies over the range $[\frac{1}{2}, 1]$ and then take a line that is completely within this intersection. A similar statement can be made about the ($\frac{1}{4}$ and $\frac{1}{2}$) region but unfortunately the resulting triangular wedges are not yet wide enough for our problem. It is clear that if we let u vary over a smaller range, we shall have wider overlap regions. Thus partitioning the u-range into three subranges, namely, $I_1 = [\frac{1}{2}, \frac{5}{8})$, $I_2 = [\frac{5}{8}, \frac{3}{4})$, and $I_3 = [\frac{3}{4}, 1]$ works well. It is clear that from the given values of c_k and b_k it is simple to determine the subrange for root u with shift, add, and comparison operations only. For example,

$$c_k - \tfrac{1}{2} b_k \geq \tfrac{1}{4}$$

and

$$c_k - \tfrac{5}{8} b_k < \tfrac{25}{64} \;=>\; u \in I_1.$$

Now we ask for three selection procedures for these three subranges of u. First we discuss the case of subrange $I_1 = [\frac{1}{2}, \frac{5}{8})$. The ($\frac{1}{2}$ and 1) overlap region is given by

$$0.485\,(b_{k-i} + \tfrac{5}{8}) \leq c_{k-i} \leq 0.72\,(b_{k-i} + \tfrac{1}{2}).$$

Similarly, the ($\frac{1}{4}$ and $\frac{1}{2}$) overlap region is given by

$$0.553\,(b_{k-i} + \tfrac{5}{8}) \leq c_{k-i} \leq 1.12\,(b_{k-i} + \tfrac{1}{2}).$$

We show these regions on the (c_{k-i}, b_{k-i}) plane, in Fig. 1.

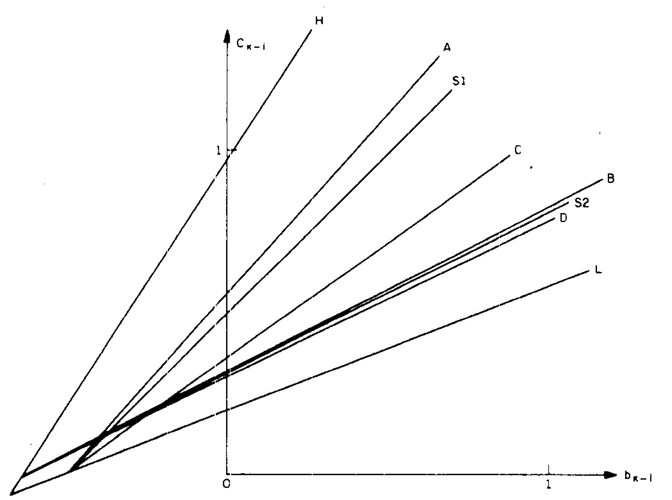


Fig. 1. The overlap regions.

The upper and the lower bounds of the ($\frac{1}{4}$ and $\frac{1}{2}$) region are labeled A and B, respectively, and those for the ($\frac{1}{2}$ and 1) region are labeled C and D. We also show the greatest upper bound $c_{k-i} = 1.56\,(b_{k-i} + \frac{5}{8})$ as line H and the least lower bound $c_{k-i} = 0.39\,(b_{k-i} + \frac{1}{2})$ as line L. We also draw two selection lines $S1$ and $S2$, which are $c_{k-i} = b_{k-i} + \frac{1}{2}$ and $c_{k-i} = \frac{1}{2}\,b_{k-i} + \frac{5}{16}$, respectively. Notice that the coefficients in these lines are chosen to be "simple" binary numbers. For any point in the region enclosed by line H and $S1$ we choose $q_{i+1} = \frac{1}{4}$. For any point between $S2$ and L, we choose $q_{i+1} = 1$ and otherwise we choose $q_{i+1} = \frac{1}{2}$. Notice that with these rules our choice could be erroneous in certain regions. Where this happens, we call these regions the forbidden regions. The quadrilateral enclosed by lines $H, B, S1$, and L is the ($\frac{1}{4}$ and $\frac{1}{2}$) forbidden region and the quadrilateral enclosed by lines H, $S2$, C, and L is the ($\frac{1}{2}$ and 1) forbidden region. We have to make sure that for no value of i, the point (c_{k-i}, b_{k-i}) lies in one of these regions. A proof of this fact can be found in [5]. A similar treatment can be given to the other two subranges I_2 and I_3. For the subrange I_2, the selection lines S_1 and S_2 are, $c_{k-i} = b_{k-i} + \frac{5}{8}$ and $c_{k-i} = \frac{1}{2}\,b_{k-i} + \frac{3}{8}$, respectively. For the subrange I_3, the selection lines $S1$ and $S2$ are $c_{k-i} = b_{k-i} + \frac{3}{4}$ and $c_{k-i} = \frac{1}{2}\,b_{k-i} + \frac{1}{2}$, respectively.

Although this general selection procedure is valid for all $i \geq 0$, we want to use a special procedure for $i = 0$ so that when we make tests for the subrange determination, we also find q_1 on the basis of the same tests.

For $i = 0$, $f_i = u$. Then from our previous analysis, we have, $0.485 \leq u \leq 1.124$ then $q_1 = \frac{1}{2}$ and $0.39 \leq u \leq 0.72$ then $q_1 = 1$. Thus we can choose $q_1 = 1$ for all $u \in I_1$ and $q_1 = \frac{1}{2}$ for all $u \in I_2$ or I_3.

We now give the complete Algorithm A.

Step A0–[Check]: If $b_k < 0$ then exit, no solution; otherwise if $(c_k - \frac{1}{2} b_k) < \frac{1}{4}$ or if $(c_k - b_k > 1)$ then exit, no solution.

Step A1–[Subrange]: If $c_k - \frac{5}{8}\,b_k < \frac{25}{64}$ then set $q_1 \leftarrow 1$, $K1 \leftarrow \frac{1}{2}$, $K2 \leftarrow \frac{5}{16}$ and go to Step A2; otherwise set $q_1 \leftarrow \frac{1}{2}$, if $c_k - \frac{3}{4}\,b_k < \frac{9}{16}$ then set $K1 \leftarrow \frac{5}{8}$, $K2 \leftarrow \frac{3}{8}$ and go to Step A2; otherwise set $K1 \leftarrow \frac{3}{4}$, $K2 \leftarrow \frac{1}{2}$.

Step A2–[Initialize]: Set $P_0 \leftarrow 0, Q_0 \leftarrow P_1 \leftarrow 1, Q_1 \leftarrow q_1$; set $b_{k-1} \leftarrow q_1 c_k, c_{k-1} \leftarrow 1 + q_1(b_k - b_{k-1})$; set $i \leftarrow 2$.

Step A3–[Selection]: If $c_{k-i+1} > (b_{k-i+1} + K1)$ then set $q_i \leftarrow \frac{1}{4}$ and go to Step A4; otherwise if $c_{k-i+1} \leqslant \frac{1}{2} b_{k-i+1} + K2$ then set $q_i \leftarrow 1$ and go to Step A4; otherwise set $q_i \leftarrow \frac{1}{2}$.

Step A4–[Advance]: Set

$$
\begin{aligned}
b_{k-i} &\leftarrow q_i c_{k-i+1} - q_{i-1} c_{k-i+2} + b_{k-i+2} \\
c_{k-i} &\leftarrow q_i (b_{k-i+1} - b_{k-i}) + c_{k-i+2} \\
P_i &\leftarrow q_i P_{i-1} + P_{i-2} \\
Q_i &\leftarrow q_i Q_{i-1} + Q_{i-2} \\
i &\leftarrow i + 1.
\end{aligned}
$$

Step A5–[Loop Test]: If $i \leqslant i_{max}$ then go to Step A3;

Step A6–[Final]: u (=ROOT$_1$) $\leftarrow P_i/Q_i, v \leftarrow b_k + u$.

Note: The value of i_{max} will be decided by the machine precision in case this algorithm is implemented in hardware. If this algorithm is implemented in software, however, the value of i_{max} will be decided by the allowable error.

VI. Recent Related Work

In the preceding sections, the discovery of the first continued fraction algorithm and its method of application have been described. In this regard, the exposition is historically ordered. The purpose of this section is to describe briefly the results of more recent research.

A study of the derivation of the quadratic algorithm of (7) has indicated that the requirement that $p_i = 1$ for all i is unnecessary [3]. Equations (7) then become

$$
\begin{aligned}
b_{k-n} &= \frac{q_n}{p_n} c_{k-n+1} - \frac{q_{n-1}}{p_{n-1}} c_{k-n+2} + b_{k-n+2} \\
c_{k-n} &= -q_n b_{k-n} + q_n b_{k-n+1} + \frac{p_n}{p_{n-1}} c_{k-n+2}. \qquad (15)
\end{aligned}
$$

Selection rules for the digit sets $p_i \in \{\frac{1}{2}, 1\}$ and $q_i \in \{\frac{1}{2}, 1\}$ have been determined.

In [4], it is shown that the Riccati equation

$$y' + ay^2 + by + c = 0 \qquad (16)$$

leads to relatively simple recursions if the partial numerators p_i and partial denominators q_i of the associated continued fraction are simple in the binary sense. Since both $\tan x$ and e^x satisfy the Riccati equation for particular choices of a, b, and c, there is some hope that useful continued fraction algorithms for these functions can be found. Attempts to find selection rules for $\tan x$ have thus far been unsuccessful and have not been attempted for the exponential.

The derivation of the recursion relations for the Riccati equation suggested a similar derivation for the quadratic equation and led to a second set of recursion relations for the quadratic. The special selection procedure for $i = 0$ described in Section V suffices for selection rules for this second quadratic algorithm. Recursion relations for higher order polynomials can also be found by this method.

VII. Conclusions

It should be emphasized that the primary purpose of this paper is to point out that hardware construction can be based on representations of numbers other than positional notation. It seems quite clear that the use of continued products yields a useful set of algorithms that can share the same hardware in a feasible and practical manner using current technology.

The discussion of continued fractions presented here is a case study of the problems that arise when a different representation of numbers is proposed. The research on the use of continued fractions is incomplete; the results obtained thus far do not justify hardware construction based on continued fractions.

It seems appropriate, therefore, to conclude with a list of questions for future research. These include the following.

1) Can the set of algorithms soluble with continued products be extended?

2) How can the set of algorithms based on the use of continued fractions be extended? Can feasible selection rules for each algorithm be found?

3) What additional representations of numbers exist? What is their potential usefulness?

References

[1] B. G. DeLugish, "A class of algorithms for automatic evaluation of certain elementary functions in a binary computer," Ph.D. dissertation, Univ. Illinois, Urbana, June 1970; also Dep. Comput. Sci. Rep. 399.

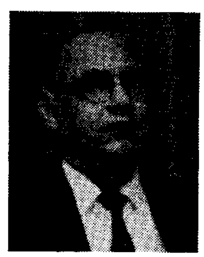
[2] M. Ercegovac, "Radix 16 division, multiplication, logarithmic and exponential algorithms based on continued product representations," M.S. thesis, Univ. Illinois, Urbana, Aug. 1972; also Dep. Comput. Sci. Rep. 541.

[3] A. Bracha, private communication.

[4] —, "Application of continued fractions for fast evaluation of certain functions on a digital computer," Dep. Comput. Sci., Univ. Illinois, Urbana, Rep. 510, Mar. 1972.

[5] K. S. Trivedi, "An algorithm for the solution of a quadratic equation using continued fractions," M.S. thesis, Univ. Illinois, Urbana, June 1972; also Dep. Comput. Sci. Rep. 525.

[6] H. Wall, *Analytic Theory of Continued Fractions.* Princeton, N.J.: Van Nostrand, 1950.



James E. Robertson (S'48–A'50–M'55) received the Ph.D. degree in electrical engineering from the University of Illinois, Urbana, in 1952.

Since 1950 he has been associated with the Department of Computer Science, University of Illinois. He was Chief Engineer for the design and construction of the high-speed computer Illiac II until December 1962. He spent the year 1963 on sabbatical leave as a Fulbright scholar at the University of Sydney, Australia. He is currently a Professor of Computer Science at the University of Illinois.

Dr. Robertson served as a member of the Program Committee for the Association for Computing Machinery National Meeting in 1957, and was Chairman of the Local Arrangements Committee for the meeting in 1958. He was a member of the ACM Council from 1958 to 1960, and was a member of the Editorial Board of the IRE Transactions on Electronic Computers.

Kishor S. Trivedi was born in Bhavnagar, India, on August 20, 1946. He received the B. Tech. (E.E.) degree from the Indian Institute of Technology, Bombay, India, in 1968 and the M.S. degree in computer science from the University of Illinois, Urbana, in 1972.

From 1968 to 1970 he was a Customer Engineer at IBM, Bombay, India. Since 1970 he has been a Graduate Research Assistant at the University of Illinois, where he is currently a candidate for the Ph.D. degree.