ROUND-OFF ERRORS IN FLOATING-POINT ADDITIONS

By

Kumiko TSUJI

(Received January 16, 1985)

Abstract

Models of algorithms of floating-point addition are designed for chopping, correctly rounding and augmentation rounding arithmetics with finite-length accumulators. The chopping arithmetic is shown to match with the simplest software, which calculates round-off errors. The effective length of digits of the accumulator is detected theoretically to this software.

1. Introduction

Softwares to calculate round-off errors in floating-point addition were introduced by Møller [4] and Kahan [2]. In the present paper, the simplest software, named by Linnainmaa [3] as Møller-Kahan method, is investigated, which will be abbreviated as M-K method.

The possibility of floating-point computation of round-off errors has been shown in [3] for chopping and correctly rounding arithmetics with so to say infinite-length accumulator. In [3], however, the round-off error is not produced correctly by the chopping arithmetic for the special case satisfying both the following conditions: (i) the signs of the augend and the addend are different and (ii) the difference of the exponents of the augend and the addend is greater than single length. Because the longest length of the round-off error is over above than the single length of the floating-point number and so overflowing digits from it are lost in the round-off error. Therefore the round-off error is obtained correctly only in the limits of single length in [3]. The chopping arithmetic of the addition in [4] fits for practical use because it assumes a finite length of digits. In [4], the addend is truncated before addition in order to fit to the single length accumulator, where two operands are rearranged such that the exponent of the augend is greater than that of the addend. But in [4], the round-off error is

obtained by other software with 7 statements which has the longer statement than M-K method with 4 statements. The purpose of the present paper is to construct the algorithm of the addition to estimate the round-off error and, at the same time, a model of the addition is selected so that the round-off error is evaluated more accurately by the simpler algorithm. The M-K method is the simplest algorithm evaluating the round-off error correctly hitherto, which is expressed in terms of single length floating-point arithmetic. Therefore the M-K method will be proper to be investigated for several models of the addition, whose algorithm is formulated mathematically. First, M-K method is examined about the algorithm of the addition in [4]. The M-K method is successful, when the addition has the following property: if the actual sum is of single length for the pairs of the floating-point numbers whose exponents are equal or differ by one, then the floating-point sum coincides with the actual sum. But the single length accumulator in [4] has not this property and it lacks the continuity of addition for the pairs of the different exponents. Then the single length accumulator is insufficient in length matching with M-K method, and the estimation of the round-off error is not obtained from the leading digit. Therefore finite-length accumulators longer than single length are investigated by chopping arithmetics.

The imperfection arising in the special case ((i) and (ii)) in [3], finds a way out to the use of (single length + 1)-digits in the accumulator. Because even in our truncation of the addend using the (single length + s)-digits ($s \geq 2$) in the accumulator, the longest length of the round-off error becomes (single length + $s - 1$)-digits. The longest length increases linearly with the increment of the length of the accumulator. Moreover, as the s increases, the unsuccessful range of the pairs of inputs becomes larger, in which the round-off error is not computed correctly by M-K method. Then the estimation becomes more inferior as s increases.

If s equals to 1, the longest length of the round-off error is just single length. Further the floating-point sum is expressed as the augend and the round-off error is expressed as the addend. This algorithm is essential to the property of round-off error not overflowing than single length digits. Therefore the simple algorithm is recommended to put out the augend as the floating-point sum and the addend as the round-off error for the special case ((i) and (ii)) of [3].

On the other hand, for the general pairs not included in the special case, the round-off errors are obtained correctly for our any finite-length accumulators ($s \geq 1$) similar to [3] for $s = +\infty$. Therefore if this simple algorithm ($s = 1$) is inserted to the special case, then the round-off errors are also obtained correctly

for every cases including the special case by using the infinite-length or our finite-length ($s \geq 1$) accumulators.

Besides, the correctly rounding arithmetic and the augmentation rounding arithmetic are examined. In [3], it has been shown that for the correctly rounding arithmetic, the M-K method does not work well in some special range of pairs of positive inputs and so the round-off error is not estimated from the leading digit while in chopping arithmetic the estimation is accurate only within the single length for any pairs. Even in our finite-length accumulators, the correctly rounding arithmetic and the augmentation rounding arithmetic do not match with M-K method independently of the length of accumulators, since the accuracy of the estimation of the round-off errors is insufficient as that in the infinite-length accumulator.

2. Definitions and notations

Let a number x be expressed as a finite power series of an integer b in the decreasing order:

$$(2.1) \quad x = \pm \sum_{i=j}^k x\{i\}b^i \quad \text{for } j \geq k, \text{ with } 0 \leq x\{i\} \leq b-1.$$

x is denoted as $x[j, k]$. If $j=k$, then x is denoted as $x[j]$. The place of digit $\pm x\{i\}b^i$ in x is called an " i -th place". For the nonzero real number $x[j, k]$, let $F(x)$ denote the place of the leading digit of x and $L(x)$ the place of the last nonzero digit of x . We put $K(x) = F(x) - L(x) + 1$. If $K(x) \leq n$, then x is called an " n -length number". We shall define one so-called normalized floating-point number. We put $e(x) = F(x) + 1$. An integer b such that $b \geq 2$ is chosen as a base, and a length t is fixed to all floating-point numbers. A certain range of integer exponents $e(x)$ is also specified ($-m \leq e(x) \leq M$, $m, M > 0$). A t -length number x with the exponent of the specified range is called the normalized floating-point number. Any nonzero floating-point number has the form $x = MT(x)b^{e(x)}$, where $MT(x)$ is given by $x/b^{e(x)}$ and expressed as $\pm \sum_{i=-1}^{t-1} x(i)b^i$ with $0 \leq x(i) \leq b-1$. The $MT(x)$ is called the mantissa. Further we put $l(x) = L(MT(x))$. In the present paper, the range of exponents is assumed $-\infty < e(x) < \infty$, but the usability of the softwares is discussed in the range where no exponent underflow or overflow occurs.

A new chopping operator is defined as follows to find the cancellation digits and the carry digit in M-K method. Let $C(x[j, k], i)$ be an operator which discards the digits less than $(i+1)$ -place for $j \geq i \geq k$:

$$(2.2) \quad C(x[j, k], i) = x[j, i+1].$$

The “chopping to s -length” $fl(\cdot, s)$ is defined as

$$fl(x, s) = C(x[F(x), L(x)], F(x) - s) \text{ i.e.}$$

$$fl(x, s) = x[F(x), F(x) - s + 1].$$

A “chopping error” is defined as $c(x, s) = x - fl(x, s)$. The floating-point additions are denoted as $u \oplus v$ for two normalized floating-point numbers u and v . The “round-off error” is defined as $r = (u + v) - u \oplus v$ for the addition $\oplus$. The various floating-point additions will be designed in sections 3 and 5. The operator C has the following property.

LEMMA 2.1. For the positive numbers $x[j, k]$ and $y[p, q]$, the operator C satisfies

$$(2.3) \quad C(x[j, k] + y[p, q], k) = x[j, k+1] + y[p, k+1] + C(x[k] + y[k], k),$$

where $p \geq k \geq q$, and the latter $C(\cdot, \cdot)$ in (2.3) is

$$C(x[k] + y[k], k) = b^{k+1} \quad \text{if } x[k] + y[k] \geq b^{k+1}$$

$$C(x[k] + y[k], k) = 0 \quad \text{if } x[k] + y[k] < b^{k+1}.$$

For the positive numbers $x[-1]$ and $y[-1]$, we put

$$(2.4) \quad z = x[-1] + y[-1] - C(x[-1] + y[-1], -1).$$

Then z is 0 or a positive 1-length number expressed as $z[-1]$.

PROOF. (2.3) is proved clearly. If $2 - 2b^{-1} \geq x[-1] + y[-1] \geq 1$ in (2.4), then $C = 1$ and $1 - 2b^{-1} \geq z \geq 0$. If $x[-1] + y[-1] < 1$, then $C = 0$ and $1 > z > 0$.

Q. E. D.

The value $C(x[k] + y[k], k)$ expresses the carry digit transformed from k -place to $(k+1)$ -place, since $C(x[k] + y[k], k)$ takes the value b^{k+1} if $x[k] + y[k] \geq b^{k+1}$ and 0 if $x[k] + y[k] < b^{k+1}$. The value $x[k] + y[k] - C(x[k] + y[k], k)$ expresses the k -place digit in $x[k] + y[k]$, since the carry digits are cancelled. By (2.3), the operator C is possible to move down to the k -place, where the chopping occurs.

3. Chopping arithmetic with finite-length accumulators

The round-off error r is rewritten as

$$r = -\{(u \oplus v - u) - v\}$$

i.e. r is calculated as

$$w = u \oplus v, \tilde{v} = w - u, r = v - \tilde{v}.$$

An estimation w' of the round-off error r is calculated by

$$w = u \oplus v, \quad v' = w \ominus u, \quad w' = v \ominus v',$$

replacing actual subtractions “ $-$ ” by the floating-point subtractions “ $\ominus$ ”, when $|u| \geq |v|$. Here $u \ominus v$ is defined as $u \oplus (-v)$. When $|v| \geq |u|$, the estimation w' is calculated by

$$w = u \oplus v, \quad v' = w \ominus v, \quad w' = v \ominus v'.$$

We call the above two procedures of w' , “*M-K method*”. The major premise (P) is inserted without notice in the following theorems: (P) u and v are two normalized floating-point numbers such that $u \geq |v|$, w' is the estimation of the round-off error obtained by M-K method, mantissa length t satisfies $t \geq 3$ and base b satisfies $b \geq 2$. The first assumption is not essential because if (u, v) does not satisfy $u \geq |v|$, then theorems hold quite the same way replacing u and v .

The following theorem 3.1 refers to the special range (O): (O) denotes the range of the pairs (u, v) satisfying both $v < 0$ and $t + s - 1 \geq e(u) - e(v) \geq t + 1$. In (O), let the addition be designed by the chopping arithmetic using $(t + s)$ -digits in the accumulator:

$$(3.1) \quad u \oplus v = fl(u + fl(v, e(v) - e(u) + t + s), t).$$

Theorem 3.1 shows that the longest length of the round-off error is $t + s - 1$, and thus $r \neq w'$ in general even though the first t -digits of r equal to w' . Because the remaining $(s - 1)$ -digits of r are cut by M-K method. Therefore t -digits memory of w' is insufficient to store the round-off error. Moreover this unsuccessful range (O) of inputs becomes larger as s increases.

THEOREM 3.1. *Let the addition be designed as (3.1) by the accumulator of $(t + s)$ -length in (O). Then the round-off error is*

$$(3.2) \quad r = b^{e(w)-t} + v$$

and the longest length of the round-off error is $(t+s-1)$ -digits.

If the length of r is longer than t i.e. $(O\alpha) e(w)-t-l(v)-e(v)>t$, then

$$(3.3) \quad r = w' + c(r, t)$$

where $c(r, t) = b^{e(w)-2t} + \sum_{i=j}^{l(v)} v(i)b^{i+e(v)}$ with $j = e(w)-2t-1-e(v)$. If the length of r is shorter than $t+1$ i.e. $(O\beta) e(w)-t-l(w)-e(v) \leq t$, then $r = w'$. Further $u+v$ is approximated by $w+w'$ up to $2t$ -digits correctly:

$$(3.4) \quad u+v = w+w' + c(u+v, 2t), \quad \text{with } c(u+v, 2t) = c(r, t).$$

Theorem 3.1 is proved from the following lemmas 3.1 and 3.2. The theory holds also, even if u, v and w are shifted to the same direction by the same digits, and so it is convenient to express w as an integer of t -digits. For this shift, we use a transformation $X = xb^{t-1-F(w)}$ for any number x . Note that $w \geq 0$ holds always since $u > |v|$, w turns into a t -length positive integer W with $F(W) = t-1$ and $L(W) = L(w) + t - 1 - F(w)$ for $w \neq 0$. Here $L(W) \geq 0$ from $L(w) \geq F(w) - t + 1$. The case $w=0$ occurs when $u = -v$ or $u=v=0$, and M - K method is successful clearly. Small letters x, u, v, w, w' and v' are transformed into capital letters respectively by this transformation.

LEMMA 3.1. Assume that the following conditions are satisfied:

$$(3.5) \quad \text{either } |F(W) - F(U)| \leq 1, \text{ or } (3.5)' \quad W < U \text{ and } L(W) \geq F(U) - s - t + 1$$

$$(3.6) \quad F(W-U) - L(W-U) + 1 \leq t$$

$$(3.7) \quad \text{either } |F(V') - F(V)| \leq 1, \text{ or}$$

$$(3.7)' \quad |V| < |V'| \quad \text{and} \quad L(V) \geq F(V') - s - t + 1$$

$$(3.8) \quad F(R) - L(R) + 1 \leq t.$$

Then $W' = R$, where $R = U + V - W$.

PROOF. Let (3.5) be satisfied and $W > U (> 0)$, then $F(W) - F(U) \leq 1$ and

$$\begin{aligned} W \ominus U &= f(W + f(-U, F(U) - F(W) + t + s), t) \\ &= f(W - U[F(U), F(W) - t + 1 - s], t) = f(W - U, t). \end{aligned}$$

Because $s \geq 1$ and $F(U) - L(U) + 1 \leq t$; so $F(W) - t + 1 - s \leq F(W) - t \leq F(U) +$

$1-t \leq L(U)$ and this means $U[F(U), F(W)-t+1-s]=U$. For the case $U \geq W \geq 0$, it holds that $F(U)-F(W) \leq 1$ and

$$W \ominus U = -fl(U + fl(-W, F(W) - F(U) + t + s), t) = -fl(U - W, t) = fl(W - U, t).$$

The condition (3.5)' says that

$$fl(-W, F(W) - F(U) + t + s) = -W.$$

Next, (3.6) means merely that $W - U$ is of t -length at most i.e. $fl(W - U, t) = W - U$. Thus

$$V' = W \ominus U = W - U.$$

(3.7) is the similar condition to (3.5) and so

$$V \ominus V' = fl(V - V', t) = fl(U + V - W, t) = fl(R, t) \text{ i.e. } W' = fl(R, t).$$

Thus by the last condition (3.8), $W' = R$ is obtained.

Q. E. D.

Let A be the accumulator value for the calculation of $U \oplus V$, i.e. $A = U + fl(V, m)$, where $m = t + s + F(V) - F(U)$. Then the round-off error is expressed as

$$\begin{aligned} R &= U + V - (U \oplus V) = U + V - A + \{A - fl(A, t)\} \\ &= V - fl(V, m) + c(A, t) = c(V, m) + c(A, t). \end{aligned}$$

This means that R consists of two chopping errors: the error $c(V, m)$ occurs in chopping of augend V and the error $c(A, t)$ occurs in chopping of the accumulator value A .

Lemma 3.2 shows that the only final (3.8) is not satisfied for some pairs (u, v) in (O), whereas (3.5)~(3.7) are satisfied.

LEMMA 3.2. Assume $s \geq 2$. In (O), (u, v) are classified into the following cases: (Oa) $F(W) = F(U)$, (Oa α) $F(W) - L(V) \leq 2t - 1$, (Oa β) $F(W) - L(V) \geq 2t$, (Ob) $F(W) = F(U) - 1$, (Ob α) $F(W) - L(V) \leq 2t - 1$, (Ob β) $F(W) - L(V) \geq 2t$.

The longest length of the round-off error in (O) is $(t + s - 1)$ -length and it arises in case (Oa). Further in (O),

$$(3.9) \quad R = 1 + V, \quad (3.10) \quad R = W' + c(R, t), \quad (3.11) \quad c(R, t) = c(U + V, 2t) = V[-t - 1, L(V)] + b^{-t} - C(b^{-t} + V[-t - 1, L(V)], -t - 1).$$

PROOF. (Oa) (3.5) holds by $F(W) = F(U)$. We have $A = U[t - 1, 0] +$

$V[F(V), -s]$ with $-2 \geq F(V) \geq -s$ and $W = U[t-1, 0] - 1$, since $V[-1, -s] \neq 0$. (3.6) holds, since $W - U = -1$. The round-off error R is expressed as (3.9), where $F(R) = -1$ and $L(R) = L(V) \geq F(V) - t + 1 \geq -s - t + 1$. Therefore the longest length of R is $t + s - 1$. (3.7)' is satisfied, since $V' = -1$, $F(V') = 0$ and $L(V) \geq F(V) - t + 1 \geq F(V') - s - t + 1$. Therefore $W' = f(R, t)$ and so (3.10) holds. Further

$$(3.12) \quad \begin{aligned} W' &= f(1 + V, t) \\ &= 1 + V[F(V), -t] - b^{-t} + C(b^{-t} + V[-t-1, L(V)], -t-1). \end{aligned}$$

If (Oa α) holds, then $L(V) \geq -t$. Therefore $c(R, t) = 0$ and $R = W'$. If (Oa β) holds, then $L(V) \leq -t-1$ and $c(R, t) \neq 0$. On the other hand, $F(U+V) = F(A) = t-1$ by $F(V) \leq -2$. Therefore $c(U+V, 2t) = c(c(U+V, t), t) = c(1+V, t) = c(R, t)$, since $f(U+V, t) = U[t-1, 0] - 1 + C((1+V)[-1, L(V)], -1)$ i.e. $c(U+V, t) = 1+V$. (Ob) (3.5) holds from $F(W) = F(U) - 1$. We have $A = U[t, 1] + V[F(V), -s+1]$ with $-s+1 \leq F(V) \leq -1$ and so $U = b^t$. W is calculated as $W = b^t - 1$ by $V[-1, -s+1] \neq 0$. (3.6) holds by $W - U = -1$. Further $R = 1 + V$, where $F(R) \leq -1$ and $L(R) = L(V) \geq F(V) - t + 1 \geq -s - t + 2$. Therefore the longest length of R is $t + s - 2$. (3.7)' follows, since $F(V') = 0$ and $L(V) \geq -s - t + 2$. The remaining part is proved in the same way as the case (Oa). Q.E.D.

The conditions (Oa α) and (Ob α) reduce to the condition (O α) by $L(v) = l(v) + e(v)$. Further (3.9), (3.10) and (3.11) reduce to (3.2), (3.3) and (3.4) returning R, U, V and W' to the former states r, u, v and w' . Therefore theorem 3.1 is proved.

Now the round-off error is obtained correctly by M-K method in the range such that (ia) $e(u) - e(v) \geq t+1$ for $v > 0$ and (ib) $e(u) - e(v) \geq t+s$ ($s \geq 1$) for $v < 0$, since the sum becomes addend u itself and round-off error becomes augend v itself:

$$(3.13) \quad u \oplus v = u \quad \text{and so} \quad r = v.$$

The following theorem 3.2 will show that the round-off error is estimated correctly in the range such that $e(u) - e(v) \leq t$ for $s \geq 1$. If $s=1$, i.e. the length of accumulator is $(t+1)$ -digits, then range (O) disappears, and $w' = r$ for all pairs of (u, v) . If $s \geq 2$, then the round-off error is not obtained correctly in range (O). It is recommended to insert a simple algorithm (3.13) in range (O) for other accumulators ($t \geq s \geq 2$) too.

For $v > 0$, chopping v to $t+s+e(v) - e(u)$ digits before addition does not affect

the value $u \oplus v$ for any $s \geq 0$ in (3.1):

$$\begin{aligned} fl(u + fl(v, t + s + e(v) - e(u)), t) \\ = fl(u + fl(v, t + e(v) - e(u)), t) \quad \text{for any } s \geq 0. \end{aligned}$$

Then (3.1) is reduced to

$$(3.14) \quad u \oplus v = fl(u + fl(v, t + e(v) - e(u)), t) \quad \text{for } v > 0.$$

For $v > 0$, (single length + 1)-accumulator is needed, since the sum $u \oplus v$ takes a figure up one place at most than u .

THEOREM 3.2. *Let the addition be designed as (3.14) by the (single length + 1)-accumulator for such pairs $v > 0$ and $e(u) - e(v) \leq t$, and designed as (3.1) by $(t + s)$ -digit-length accumulator for such pairs $v < 0$ and $e(u) - e(v) \leq t$. Then $w' = r$ for any s satisfying $t \geq s \geq 1$.*

All pairs of u and v are classified into the following cases. The next (ia) and (ib) have been treated in (3.13).

- (ia) $v > 0$ and $F(u) - F(v) \geq t$. (ib) $v < 0$ and $F(u) - F(v) \geq t + s$.
- (ii) $v > 0$ and $F(u) - F(v) \leq t - 1$,
 - (iia) $F(w) = F(u) + 1$, (iib) $F(w) = F(u)$.
- (iii) $v < 0$ and $F(u) - F(v) \leq s$,
 - (iiia) $F(v) < F(w)$, (iiib) $F(v) \geq F(w)$.

The next (iv) is deleted if $s = t$.

- (iv) $v < 0$ and $t \geq F(u) - F(v) \geq s + 1$,
 - (iva) $F(w) = F(u)$,
 - (iva α) $v[m_1, m_2] \neq 0$, (iva β) $v[m_1, m_2] = 0$,
 - (ivb) $F(w) = F(u) - 1$,
 - (ivb α) $v[m_1, m_3] \neq 0$, (ivb β) $v[m_1, m_3] = 0$.

Here $m_1 = F(w) - t$, $m_2 = F(w) - t - s + 1$, $m_3 = F(w) - t - s + 2$. The next (O) has been treated already as the special case in theorem 3.1. (O) is deleted if $s = 1$.

- (O) $v < 0$ and $t + s - 1 \geq F(u) - F(v) \geq t + 1$, (Oa) $F(w) = F(u)$,
 - (Oa α) $F(w) - L(v) \leq 2t - 1$, (Oa β) $F(w) - L(v) \geq 2t$, (Ob) $F(w) = F(u) - 1$,
 - (Ob α) $F(w) - L(v) \leq 2t - 1$, (Ob β) $F(w) - L(v) \geq 2t$.

Theorem 3.2 treats the cases (ii) ~ (iv) and is proved by the following lemmas 3.3 and 3.4.

LEMMA 3.3. We have $W' = R$ in the cases (iia), (iiia), (ivax) and (ivbx).

PROOF. (3.5), (3.6), (3.7) and (3.8) are checked in each case.

(iia) (3.5) holds from $F(W) = F(U) + 1$. Since $F(W)$ is fixed to equal to $t - 1$, $F(U)$ becomes $t - 2$. A is calculated as $A = U[t - 2, -1] + V[F(V), -1]$, since U and V are positive. W is given as $W = fl(A, t) = U[t - 2, 0] + V[F(V), 0] + C(U[-1] + V[-1], -1)$, since $F(A) = F(W) = t - 1$. $W - U$ is expressed as $W - U = V[F(V), 0] - U[-1] + C(U[-1] + V[-1], -1)$. $F(W - U) = F(V)$ if $C = 1$, since $U[-1] \neq 0$. $F(W - U) = F(V)$ or $F(V) - 1$ if $C = 0$. Therefore (3.7) holds. In both cases, $F(W - U) \leq t - 2$ and $L(W - U) \geq -1$, and so (3.6) holds. The round-off error R is obtained as $R = c(V, F(V) + 2) + c(A, t) = V[-2, L(V)] + U[-1] + V[-1] - C(U[-1] + V[-1], -1) = U[-1] + V[-1, L(V)] - C(U[-1] + V[-1], -1)$. We have $F(R) \leq -1$ from lemma 2.1 and $L(R) = \min. (-1, L(V)) \geq -t$, since $F(U) - F(V) \leq t - 1$. Therefore (3.8) is satisfied.

(iii) $c(V, F(V) - F(U) + t + s) = 0$ since $F(U) - F(V) \leq s$. Therefore $A = U + V$ and $R = c(A, t)$.

(iiia) If $F(V) = F(U) - 1$, then $F(W) = F(U)$ and $F(V) = t - 2$ since $F(V) < F(W)$ and $F(W) \leq F(U)$ by $V < 0$. We have $W = U[t - 1, 0] + V[t - 2, 0] - 1 + C(1 + V[-1], -1)$ and $W - U = V[t - 2, 0] - 1 + C(1 + V[-1], -1)$. If $V[-1] \neq 0$, then $F(W - U) = F(V)$ or $F(V) + 1$. If $V[-1] = 0$, then $F(W - U) = F(V)$. Therefore (3.6) and (3.7) hold. The round-off error is obtained as $R = V[-1] + 1 - C(1 + V[-1], -1)$ and finally (3.8) is reduced from $F(R) = L(R) = -1$. If $F(V) \leq F(U) - 2$, then $F(W) \geq F(U) - 1$ and $F(U) = t$ or $t - 1$ from $F(U) \geq F(W)$. For $F(U) = t$, we have $W = U[t, 1] + V[F(V), 0] - 1 + C(1 + V[-1, L(V)], -1)$ and $W - U = V[F(V), 0] - 1 + C(1 + V[-1, L(V)], -1)$. If $C = 0$, then $F(W - U) = F(V)$ or $F(V) + 1$. If $C = 1$, then $F(W - U) = F(V)$, where $t - 2 \geq F(V)$. Therefore (3.7) holds. (3.6) holds from $L(W - U) \geq 0$. The round-off error is obtained as $R = V[-1, L(V)] + 1 - C(1 + V[-1, L(V)], -1)$ with $F(R) \leq -1$. Finally $F(V) \geq F(U) - s = t - s \geq 0$ and (3.8) holds from $L(R) = L(V) \geq -t + 1$. If $F(U) = t - 1$, then $W = U[t - 1, 0] + V[F(V), 0] - 1 + C(1 + V[-1, L(V)], -1)$ and $W - U = V[F(V), 0] - 1 + C(1 + V[-1, L(V)], -1)$ with $F(W - U) = F(V)$ or $F(V) + 1$, where $t - 3 \geq F(V) \geq t - 1 - s \geq -1$. Therefore (3.6) and (3.7) hold. Finally $R = V[-1, L(V)] + 1 - C(1 + V[-1, L(V)], -1)$ with $F(R) \leq -1$ and $L(R) \geq -t$.

(ivax) (3.5) holds by $F(W) = F(U)$. We have $A = U[t - 1, 0] + V[F(V), -s]$, $W = U[t - 1, 0] + V[F(V), 0] - 1$ from $V[-1, -s] \neq 0$, and $W - U = V[F(V), 0] - 1$. Further $F(W - U) = F(V)$ or $F(V) + 1$ and $L(W - U) \geq 0$, where $t - s - 2 \geq F(V) \geq$

-1 , and so (3.6) and (3.7) hold. (3.8) is satisfied, since $R = c(V, F(V) + s + 1) + c(A, t) = V[-s-1, L(V)] + \{1 + V[-1, -s]\} = 1 + V[-1, L(V)]$ with $F(R) \leq -1$ and $L(R) \geq -t$.

In (ivb α), we assume that $s \geq 2$, since $m_3 > m_1$ and $v[m_1, m_3]$ is empty if $s = 1$. Its proof is similar to (iva α). Q. E. D.

LEMMA 3.4. *We have $W' = R$ in the cases (iib), (iiib), (iva β), (ivb) with $s = 1$ and (ivb β) with $s \geq 2$.*

PROOF. We shall show that (3.6), (3.7) and (3.8) are satisfied if (3.5) and

$$(3.15) \quad L(A) \geq 0.$$

If (3.15) holds, then A is a t -digit number, since $F(A) = F(W) = t - 1$. We have $A = W$ and so $W - U = A - U = fl(V, m)$. Therefore $V' = W - U$ by (3.5). (3.7) is satisfied since $F(V') = F(V)$. (3.8) is satisfied, because the round-off error arises only from the chopping error of $V: R = c(V, m)$. If (3.5)' and (3.15) hold, then (3.6) ~ (3.8) are also satisfied. (3.5) and (3.15) (or (3.5)' and (3.15)) will be checked by expressing the accumulator value.

(iib) (3.5) holds by $F(W) = F(U)$ and $A = U[t-1, 0] + V[F(V), 0]$ by definition.

(iiib) We have $L(A) = \min.(L(U), L(V)) \geq F(W) - t + 1 = 0$, since $F(V) \geq F(W)$. Further $W = U + V$ and $W < U$. (3.5)' is satisfied, since $L(W) \geq F(U) - s - t + 1$.

(iva β) (3.5) holds by $F(W) = F(U)$. We have $A = U[t-1, 0] + V[F(V), -s] = U + V[F(V), 0]$ by $V[-1, -s] = 0$.

(ivb) If $s = 1$, then $A = U[t, 1] + V[F(V), 0]$. In (ivb β), we assume $s \geq 2$, and the proof is similar to (iva β). Q. E. D.

If the length of accumulator is longer than double length ($s \geq t + 1$), the result of theorem 3.2 becomes quite the same as that of double length accumulator ($s = t$). Because the floating-point sum is independent of the length of the accumulator for $s \geq 0$ if $v > 0$, and the sum coincides with that of $s = t$ if $v < 0$: the sum is obtained by chopping the actual value $u + v$ which is stored in the accumulator if $v < 0$, since $v = fl(v, e(v) - e(u) + t + s)$ by $e(u) - e(v) \leq t$ and $s \geq t + 1$. Then it is impractical to use the longer length accumulator than double length.

4. Numerical examples

Facom OSIV/F4 M382 has the algorithm of addition designed by chopping arithmetic, which uses $(t+1)$ -digits in the accumulator. The round-off errors

are computed correctly by the M-K method in all cases. Here $b = 16$ and numbers 10~15 are denoted as A~F.

In case (ib), the round-off errors could not be obtained by infinite-length accumulator in [3]. But by our accumulator, it is obtained correctly. Here a number in () denotes the exponent.

(ia) $u = 0.15323F(9)$, $v = 0.746524(2)$, $w + w' = u + v = 0.15323F0746524(9)$, $w = 0.15323F(9)$, $v' = 0$, $r = w' = 0.746524(2)$.

(ib1) $u = 0.FE543A(9)$, $v = -0.ABCDEF(2)$, $w + w' = u + v = 0.FE5439F543211(9)$, $w = 0.FE543A(9)$, $v' = 0$, $r = w' = -0.ABCDEF(2)$.

(ib2) $u = 0.1(9)$, $v = -0.1(2)$, $w + w' = u + v = 0.FFFFFFFF(8)$, $w = 0.1(9)$, $v' = 0.1(2)$, $r = w' = -0.1(2)$.

In (ii), w is obtained by chopping the sum $u + v$ to t -digits.

(iia) $u = 0.FFF123(9)$, $v = 0.123456(7)$, $w + w' = u + v = 0.100035756(10)$, $w = 0.100035(10)$, $v' = 0.122D(7)$, $r = w' = 0.756(4)$.

(iib) $u = 0.EFF123(9)$, $v = 0.456123(7)$, $w + w' = u + v = 0.F0368423(9)$, $w = 0.F03684(9)$, $v' = 0.4561(7)$, $r = w' = 0.23(3)$.

In (iiia) and (iiib), the sum is stored in the accumulator and the round-off error arises only from the chopping of the value stored in the accumulator.

(iiia) $u = 0.FEF123(9)$, $v = -0.6789AB(8)$, $w + w' = u + v = 0.F878885(9)$, $w = 0.F87888(9)$, $v' = -0.6789B0(8)$, $r = w' = 0.5(3)$.

In (iiib), the length of the sum is t -digits and the round-off error is zero.

(iiib) $u = 0.123456$, $v = -0.112345$, $w + w' = u + v = 0.11111(-1)$, $w = 0.111110(-1)$, $v' = -0.112345$, $r = w' = 0$.

The same value is taken in (iva α) and (iva β) as input u . The v in (iva α) and v in (iva β) are different i.e. $e(u) - e(v) - 1 - t$ th digits of both v 's are different in the mantissa: $v(-3) \neq 0$ in (iva α) and $v(-3) = 0$ in (iva β 1). The remaining digits of both v 's are the same in (iva α) and (iva β 1). But w is obtained from the sum by chopping in (iva α) and by raising to a unit in (iva β 1).

(iva α) $u = 0.2000FF(8)$, $v = -0.FFFFFFFF(4)$, $w + w' = u + v = 0.1FFFFFFF0001(8)$, $w = 0.1FFFFFFF(8)$, $v' = -0.1(5)$, $r = w' = 0.1(-1)$.

In (iva β) and (ivb), the round-off error arises only from the chopping error of v chopped to $t + 1 + e(v) - e(u) = 3$ digits.

(iva β 1) $u = 0.2000FF(8)$, $v = -0.FF0FFF(4)$, $w + w' = u + v = 0.1FFFFFFF001(8)$, $w = 0.2(8)$, $v' = -0.FF(4)$, $r = w' = -0.FFF(1)$.

(iva β 2) $u = 0.1000FF(8)$, $v = -0.FF0FFF(4)$, $w + w' = u + v = 0.FFFFFFFF001(7)$, $w = 0.1(8)$, $v' = -0.FF(4)$, $r = w' = -0.FFF(1)$.

The same value is taken as input u in (iva β 3) and (iva β 1). Further $f(v, e(v) -$

$e(u)+t+1$ in (iva β 3) and that in (iva β 1) are the same value. Only chopping errors $c(v, e(v)-e(u)+t+1)$ of both v 's are different each other and $c(v, 3)=0$ in (iva β 3). Therefore $r=0$.

(iva β 3) $u=0.2000FF(8)$, $v=-0.FF(4)$, $w+w'=u+v=0.2(8)$, $w=0.2(8)$, $v'=-0.FF(4)$, $r=w'=0$.

(ivb1) $u=0.1000FF(8)$, $v=-0.FFFFFFF(4)$, $w+w'=u+v=0.FFFFFFF0001(7)$, $w=0.FFFFFFF1(7)$, $v'=-0.FFF(4)$, $r=w'=-0.FFF(1)$.

Only $c(v, 3)$ in (ivb2) and that in (ivb1) are different each other. The remaining digits of v in (ivb2) and that in (ivb1) are the same. Further the same value is taken as input u in (ivb2) and u in (ivb1). In (ivb2), $r=c(v, 3)=0$.

(ivb2) $u=0.1000FF(8)$, $v=-0.FFF(4)$, $w+w'=u+v=0.FFFFFFF1(7)$, $w=0.FFFFFFF1(7)$, $v'=-0.FFF(4)$, $r=w'=0$.

5. Comparison with various arithmetics

Correctly rounding and augmentation rounding arithmetics are shown not to match with M-K method.

Let x be a positive number of the form $x[j, k]$ with $j=F(x)$. Correctly rounding arithmetic to t -digits is defined as

$$E(x, t) = fl(x, t) \quad \text{if } x[j-t]/b^{j-t} < b/2, \text{ and}$$

$$E(x, t) = fl(x, t) + b^{j-t+1} \quad \text{if } x[j-t]/b^{j-t} \geq b/2.$$

The algorithm of addition of correctly rounding arithmetic is designed using at most $(t+s)$ -digits in the accumulator for $s \geq 1$:

$$(5.1) \quad u \oplus v = u \text{ if } v < 0 \text{ and } e(u) - e(v) \geq t + s, \text{ or } v > 0 \text{ and } e(u) - e(v) \geq t + 1,$$

$$(5.2) \quad u \oplus v = E(u + fl(v, e(v) - e(u) + t + s), t) \text{ if } v < 0 \text{ and } e(u) - e(v) \leq t + s - 1.$$

For $v > 0$, the value $u \oplus v$ does not change with the length of the accumulator if it is longer than t . Therefore only $(t+1)$ -digits are used in the accumulator. In case satisfying both $v > 0$ and $e(u) - e(v) \leq t$, we put $A' = u + fl(v, e(v) - e(u) + t)$. If A' takes a figure up than u i.e. $e(A') = e(u) + 1$, then the number A' of $(t+1)$ -digits is stored into the accumulator. If $e(A') = e(u)$, the digit of $(e(u) - t - 1)$ -th place in $u + v$ is added to A' and stored into the accumulator. The value A of the accumulator is defined as

$$A = A' \quad \text{if } e(A') = e(u) + 1 \quad \text{and}$$

$$A = A' + v(-e(v) + e(u) - t - 1)b^{e(u)-t-1} \quad \text{if } e(A') = e(u).$$

Finally

$$(5.3) \quad u \oplus v = E(A, t) \quad \text{if } v > 0 \quad \text{and} \quad e(u) - e(v) \leq t.$$

THEOREM 5.1. *Let the algorithm of correctly rounding arithmetic be designed as (5.1)~(5.3) for $s \geq 1$. Then*

$$r = \{u(-t) + v(-t)\}b^{-t+e(u)} - 2b^{-t+e(u)+1} \quad \text{i.e. } r = w' - (b - u(-t))b^{-t+e(u)},$$

if the following conditions (5.4)~(5.6) are satisfied, and $r = w'$ otherwise:

$$(5.4) \quad e(u) = e(v) \quad \text{and} \quad v > 0$$

$$(5.5) \quad u(-1) = u(-2) = \cdots = u(-t+1) = v(-1) = v(-2) = \cdots = v(-t+1) = b-1$$

$$(5.6) \quad u(-t) + v(-t) \geq 3b/2.$$

THEOREM 5.2 (*Linnainmaa's infinite-length accumulator* [3]). *Theorem 5.1 holds if "(5.1)~(5.3)" are replaced by*

$$(5.7) \quad u \oplus v = E(u + v, t),$$

i.e. $u \oplus v$ is defined using the infinite-length accumulator.

The condition (5.6) holds only if $b \geq 4$. Therefore for $b = 2$ or 3 , the round-off errors are estimated by w' correctly.

The chopping arithmetics are better than the correctly rounding arithmetics by comparing theorems 3.1 and 3.2 with theorem 5.1. Because in the case satisfying (5.4)~(5.6) for $b \geq 4$, $w - u$ becomes $(t+1)$ -digits in the calculation of v' in M-K method, and the round-off error r is not estimated by w' correctly for each s such that $s \geq 1$. The estimation w' is bad in this case, since r differs from w' in the leading digit of the $-t + e(u)$ th place.

For $x = x[j, k]$ with $j = F(x)$, the augmentation rounding arithmetic to t digits is defined in [9] by

$$D(x, t) = x \quad \text{if } x[j-t, k] = 0,$$

$$D(x, t) = fl(x, t) + b^{j-t+1} \quad \text{if } x[j-t, k] \neq 0.$$

The algorithm of the addition is designed for the augmentation rounding arithmetic using at most $(t+s)$ -digits in the accumulator ($s \geq 1$):

$$(5.8) \quad u \oplus v = u, \quad \text{if } e(u) - e(v) \geq t + s;$$

$$(5.9) \quad u \oplus v = D(u + fl(v, e(v) - e(u) + t + s), t), \\ \text{if } e(u) - e(v) \leq t + s - 1 \text{ and } v < 0;$$

$$(5.10) \quad u \oplus v = D(A, t), \quad \text{if } v > 0 \text{ and } e(u) - e(v) \leq t + s - 1.$$

Here A is given as

$$A = A' = u + fl(v, e(v) - e(u) + t + s - 1), \quad \text{if } e(A') = e(u) + 1$$

$$A = A' + v(-e(v) + e(u) - t - s)b^{e(u)-t-s}, \quad \text{if } e(A') = e(u).$$

THEOREM 5.3. *Let the addition be designed by (5.8)~(5.10) for the augmentation rounding arithmetic ($s \geq 1$). If (5.4), (5.5) and $u(-t) + v(-t) > b$ are satisfied, then*

$$r = \{u(-t) + v(-t)\}b^{-t+e(u)} - 2b^{-t+e(u)+1} \text{ i.e. } r = w' + u(-t)b^{e(u)-t}.$$

If $v > 0$, $t + s - 1 \geq e(u) - e(v) \geq t + 1$ and $e(u) - e(v) - l(v) > 2t$, then the estimation w' of the round-off error r is obtained in the accuracy of t -digits:

$$(5.11) \quad r = v - b^{e(u)-t} \text{ i.e. } w' = D(r, t) = r - \sum_{i=j}^{l(v)} v(i)b^{i+e(v)}$$

$$\text{with } j = e(u) - 2t - 1 - e(v).$$

In the other cases, $r = w'$.

If $s = 1$, then the second bad case ($v > 0$, $t + s - 1 \geq e(u) - e(v) \geq t + 1$ and $e(u) - e(v) - l(v) > 2t$) is excluded. However the M-K method does not work for the pairs satisfying (5.4), (5.5) and $u(-t) + v(-t) > b$ in the same way as the addition by the correctly rounding arithmetic, since the length of $w - u$ becomes $(t + 1)$ -digits. The correctly rounding arithmetic is better than the augmentation rounding arithmetic for $s = 1$. Because the pairs of (u, v) for which the round-off error differs from the estimation w' in the leading digit i.e. $e(u) - t$ th place, are fewer in the correctly rounding arithmetic. If $s \geq 2$, then both troublesome phenomena of the correctly rounding and chopping arithmetics arise. The second bad case is that the longest length of the round-off error is over above single length i.e. of $s + t - 1$ length in the same way as the addition by the chopping arithmetic. Therefore the longer accumulator is more inferior in the augmentation rounding arithmetic. We conclude that round-off errors are estimated most effectively by our chopping arithmetic recommended in section 2.

Theorem 5.4 shows that chopping arithmetic with single length accumulator

in [4] is far more inferior than the use of $(t+s)$ -digits ($s \geq 1$) in the accumulator. In [4], the numbers were expressed by binary representation ($b=2$). We generalize the base to any integer ($b \geq 2$) in theorem 5.4.

Let (q) denote the condition that $v > 0$, $e(u) - e(v) \leq t - 1$, $e(w) = e(u) + 1$ and $u(-t) + v(j) \geq b$, where we put $j = -e(v) + e(u) - t$.

THEOREM 5.4. *Let the addition be designed for chopping arithmetic as*

$$u \oplus v = fl(u + fl(v, e(v) - e(u) + t), t).$$

If (q) holds, $j + 1 \leq -1$ and $v(-1) = \dots = v(j + 1) = b - 1$, then

$$(5.12) \quad r = w' + u(-t)b^{-t+e(u)} + v(-t)b^{-t+e(v)},$$

where

$$(5.13) \quad r = u(-t)b^{-t+e(u)} + \sum_{i=j}^{l(v)} v(i)b^{i+e(v)} - b^{e(u)+1-t}.$$

If (q) holds and $j + 1 = 0$, then (5.12) holds, where r is given as (5.13) by putting $j = -1$. If (q) holds and $v(i) \neq b - 1$ for some i ($-1 \geq i \geq j + 1$), then

$$(5.14) \quad r = w' + u(-t)b^{-t+e(u)},$$

where r is given as (5.13). If $v > 0$, $e(u) - e(v) \leq t - 1$, $e(w) = e(u) + 1$ and $u(-t) + v(j) < b$, then (5.14) holds, where

$$(5.15) \quad r = u(-t)b^{-t+e(u)} + \sum_{i=j}^{l(v)} v(i)b^{i+e(v)}.$$

Otherwise $r = w'$, where

$$r = \sum_{i=j}^{l(v)} v(i)b^{i+e(v)}$$

In (5.13) and (5.15), the place of the leading digit in r is $-t + e(u)$ and so the leading digit of w' differs from that of r . Moreover the range of pairs (u, v) satisfying (q), $j + 1 \leq -1$ and $v(-1) = \dots = v(j + 1) = b - 1$, includes the unsuccessful range satisfying (5.4), (5.5) and $u(-t) + v(-t) > b$ for augmentation rounding arithmetic. The chopping arithmetic with single length accumulator is more inferior than the augmentation rounding arithmetic with $(t + 1)$ -length accumulator, though the round-off error is of t -length.

Theorems 5.1 and 5.3 are proved in the author's paper [6] and theorem 5.4 is proved in [7].

Acknowledgements

The author greatly acknowledges the guidance and the valuable advices of Professor Seiiti Huzino in Kyushu University.

References

1. DEKKER, T. J., *A floating-point technique for extending the available precision*, Numer. Math. **18** (1971), 224–242.
2. KAHAN, W., *Further remarks on reducing truncation errors*, CACM **8** (1965), 40.
3. LINNAINMAA, S., *Analysis of some known methods of improving the accuracy of floating-point sums*, Bit **14** (1974), 167–202.
4. MØLLER, O., *Quasi double-precision in floating-point addition*, Bit **5** (1965), 37–50.
5. SHIMIZU, T. AND OOHASHI, T., *Rounding errors in floating-point addition*, TRU Mathematics, Vol. **11**, (1975), 41–50.
6. TSUJI, K., *Round-off error in correctly rounding arithmetic*, RMC **59-04** (1984), 23 pages in Kyushu Univ..
7. TSUJI, K., *Round-off error in addition of two floating-point numbers*, RMC **59-03** (1984), 41 pages in Kyushu Univ..
8. WILKINSON, J. T., *Rounding errors in algebraic processes*, Her Majesty's Stationery Office (1963).
9. YOHE, J. M., *Roundings in floating-point arithmetic*, IEEE Transactions on Computers, Vol. C-**22**, No. 6 (1973), 577–586.

*Department of Mathematics
Kyushu University 33
Hakozaki, Fukuoka 812
Japan*