PROGRAMMER-CONTROLLED ROUNDOFF AND THE SELECTION OF A STABLE ROUNDOFF RULE

Roy A. Keir
University of Utah
Salt Lake City, Utah 84112

The author suggests that every computer with floating-point addition and subtraction should have PSW controlable roundoff facilities. Yohe's catalog should be included. There should also be a stable roundoff mode using the round-to-off or round-to-even rule based on whether the radix is divisible by four or only by two.

Drift-Free Rounding

The Banker's Rule

This paper mainly discusses the kind of rounding rule which should be provided for garden-variety floating-point additions and subtractions. While the "banker's rule" serves as a starting point for the discussion, the most important piece of background appeared in an article by Reiser and Knuth titled "EVADING THE DRIFT IN FLOATING-POINT ADDITION."[1] The discussion of the radix-dependence of the stable rounding rule appeared in an article by the author titled "SHOULD THE STABLE ROUNDING RULE BE RADIX-DEPENDENT?"[2]

Reiser and Knuth quote Scarborough's 1950 definition of unbiased roundoff, "It should be obvious to any thinking person that when a 5 is cut off, the preceding digit should be increased by 1 in only half the cases, ... Since even and odd digits occur with equal frequency, on the average, the rule that the odd digits be increased by 1 when a 5 is dropped is logically sound."

Where more than one digit is to be discarded, only the remainders that weigh exactly half a unit in the last digit to be retained should be subject to the "50% rule", as shown in the following chart. Scaling is in fractions of the last digit to be retained. Positive operands are bounded by f and (f+1), negative operands by g and (g-1).

Result Operand	Chopping S-Mag	Chopping R-comp	Rounding S-Mag	Rounding R-comp	Stable Rounding
f+1.00	f+1	f+1	f+1	f+1	f+1
f+ .99	f	f	f+1	f+1	f+1
f+ .51	f	f	f+1	f+1	f+1
f+ .50	f	f	f+1	f+1	50% rule #1
f+ .49	f	f	f	f	f
f+ .01	f	f	f	f	f
f+ .00	f	f	f	f	f
g- .00	g	g	g	g	g
g- .01	g	g-1	g	g	g
g- .49	g	g-1	g	g	g
g- .50	g	g-1	g-1	g	50% rule #2
g- .51	g	g-1	g-1	g-1	g-1
g- .99	g	g-1	g-1	g-1	g-1
g-1.00	g-1	g-1	g-1	g-1	g-1

In order that Round(-x) = -Round(x), it is necessary for "50% rule #1" to be identical to "50% rule #2".

Reiser and Knuth point out that while Scarborough's argument is not quite valid because of the logarithmic dependence of leading operand digits (of which the last retained digit is merely an extreme case), nonetheless a round-to-even or round-to-odd rule should be employed because either rule prevents continued drift in the computation;

$$x \oplus y \ominus y \oplus y \ominus y$$

They prove the following two theorems:

Theorem 1. If stable rounding is used, $(((x \oplus y) \ominus y) \oplus y) \ominus y = (x \oplus y) \ominus y$ for all floating-point numbers x and y such that the operations are defined.

Theorem 2. There exist x, y such that $((x \oplus y) \ominus y) \oplus y \neq x \oplus y$, regardless of what rounding scheme is used, when the radix is even.

(As usual, the circled operators here represent the finite-precision approximations to the real operators).

Other "Stable" Rounding Rules

At this point, one might pause to ask a number of questions, some of which require extensive experiments to answer.

(a) Are other stable rounding rules possible?

(b) Which is best among the stable rounding rules?

(c) Is stable rounding significantly better than biased rounding?

(d) Does stable rounding have any undesirable characteristics?

Some obvious possibilities for the "50% rule" in addition to round-to-even and round-to-odd are to "roll dice" (which makes calculations non-repeatable), to use a pre-arranged list of decisions having good statistical properties, or to base the rounding decision on some digit other than the last operand digit to be retained.[4]

Can any of these rules be considered to be "stable"? One should not expect any of them to be drift-free in the terms of the theorems put forth by Reiser and Knuth. On the other hand, one might consider the criterion of their theorems to be artificially narrow in the sense that it considers only absolute stability during a specific and rather rare sequence of operations.

Loosely speaking, the real-world hazard appears to be that "entrainment" might occur between the statistics of the roundoff criterion and some property of the programmed sequence of operations. Reiser and Knuth's proofs seem to depend on an entrainment of this sort to stabilize the sequence of results of an add-restore cycle endlessly repeated. One might expect other correlations to be less favorable. Suppose for example, that a "pre-arranged list" consists of a single toggle which causes rounding-up in strictly alternating occurrences of the "50% rule" case. One could surely invent a two-summation loop which could achieve unacceptable answers using this rule. With considerable effort, one should be able to concoct a cycle which (perhaps using the bias introduced by the logarithmic distribution) can lock in on the oddness of the last retained digit to cause drift even using a "stable" rounding rule! For a more severe example of entrainment, see Kahan's convergence example which is discussed below.

A speculation which seems worthy of some experimentation is that a long-period pseudo-random sequence might provide a kind of dynamic stability that is more important in real-world computing than the static stability resulting from use of a drift-free rounding rule.

Round-to-odd Versus Round-to-even

In comparing the even-ness based rules against other "50% rule" candidates, we should use the better of the two. Which is better? Two arguments have been published on this subject, and a third has been advanced in informal discussions.

(1) Scarborough's banker's rule uses round-to-even (in decimal arithmetic) to reduce error in subsequent divisions by two, a fairly common operation.

(2) The possibility of repeated rounding favors selection of round-to-odd if the radix is divisible by four, round-to-even if the radix is divisible only by two. The example below shows a case in which repeated roundoff using round-to-even does not give the nearest representable answer in radix-16 arithmetic. The calculation is ((7.780⊕800.0) ⊕ 800.0) for which the nearest four-digit answer is 1007. The registers are four digits long including any guard digits.

Action	Round to Even	Round to Odd	Nearest Representable
	7.780 800.0	7.780 800.0	7.780 800.0
x ⊕ y scale round	807.7(8) 807.7(8) 807.8	807.7(8) 807.7(8) 807.7	807.78
	807.8 800.0	807.7 800.0	807.78 800.0
x ⊕ y	(1)007.8	(1)007.7	1007.78
scale round	1007.8 1008.	1007.7 1007.	 1007.

(Example from reference 2.)

(3) Both of the above arguments are simple ones based on avoiding the generation of a last-retained-digit equal to half the radix. Professor Kahan advances a much tougher version of the first argument by showing that either rule can be treacherous when divide-by-two occurs in a convergence calculation, as often happens. In any even radix b (whether divisible by four or not) the following sequence can occur in calculation x : = (1 + x)/2

```
        1. 0    0
  +     0.(b-1)(b-1)
        1.(b-1)(b-1)
  /2     .(b-1)(b-1)(b/2)
  +     1.  0    0

        1.(b-1)(b-1)(b/2)

        ROUND-TO-ODD        ROUND-TO-EVEN

        1.(b-1)(b-1)        2. 0  0
  /2     .(b-1)(b-1)(b/2)   1. 0  0
```

Either outcome might be unattractive in a particular situation, but it is clear that round-to-odd does not give the nearest representable answer.

Recommendations

Computer Design

The preceding discussion indicates that one must balance frequency-of-occurence arguments, average benefits, and cost-of-disaster estimates when choosing a roundoff rule. I propose that neither round-to-even nor round-to-odd is acceptable for convergence calculations, which require the kind of dependable bounds on results which have been proposed by Yohe and are summarized in the following table.

For $0 < e < 1$	Upward	Downward	Inward	Outward
f + 1.000	f+1	f+1	f+1	f+1
f + e	f+1	f	f	f+1
f + 0.000	f	f	f	f
g - 0.000	g	g	g	g
g - e	g	g-1	g	g-1
g - 1.000	g-1	g-1	g-1	g-1

With considerably lower confidence I suggest that if the above processes are available for use where the convergence properties are known, the remaining garden-variety calculations should use drift-free rounding according to one of the following rules:

(1) If the radix is divisible by four, use round-to-odd in the 50% case.

(2) If the radix is divisible by 2 but not by 4, use round-to-even in the 50% case.

The possible desirability of providing a "statistical" roundoff for the 50% case requires more investigation. The well-known facility of making the entire extended result of the computation available to the program (and doing no rounding at all) seems worthy of retention, even if it were used only for diagnostic purposes.

Since a major use of Yohe's directionally-bounded rules is expected to be in mathematical subroutines and functions, it is essential that the selection of the rule to be in force during a segment of a computation be part of the testable, adjustable, automatically restored context as for example in the Program Status Word or equivalent.

Professional

In closing, I would like to suggest several activities which might be undertaken by attendees of this Symposium for the benefit of the computing field:

(1) Test the impact of stable rounding versus normal (biased) rounding in real-world calculations.

(2) Find more persuasive criteria for the selection of an unbiased rounding rule.

(3) Require of ourselves and of suppliers of arithmetic services and hardware more complete specification of the arithmetic performed, not just the words "chopped" and "rounded".

References

(1) J. F. Reiser and D. E. Knuth, "Evading the drift in floating-point addition", Info. Proc. Letters vol 3 no. 3 (January 1975) pp 84-87.

(2) R. A. Keir, "Should the Stable Rounding Rule be Radix-dependent?", Info. Proc. Letters, to be published.

(3) The processing unit of the (Bendix/CDC)(G-20/G-21) used a round-to-odd rule in radix eight. If anyone knows of studies which were performed on G-20 arithmetic, I would be grateful for a reference to such material.

(4) Roland Silver, Letter to the Editor, CACM, Dec. 1960.