FLOATING-POINT ON-LINE ARITHMETIC: ERROR ANALYSIS*

O. Watanuki and M. D. Ercegovac

UCLA Computer Science Department
University of California, Los Angeles

ABSTRACT -- The properties of redundant number system in mantissa representation are studied and the range of the redundant mantissa is derived. From the range of the mantissa and the absolute error of on-line operations, the MRRE (maximum relative representation error) is defined and analyzed for redundant floating-point numbers.

1. INTRODUCTION

Introducing the notion of quasi-normalization, FLPOL (floating-point on-line) arithmetic algorithms are presented in the associated paper [WaEr81]. In this paper we study the properties of the redundant number system in mantissa [Aviz61] and the error characteristics of FLPOL arithmetic operations.

A number of measures have been proposed to evaluate the performance of computer arithmetic. Wilkinson [Wilk63] used the maximum relative representation error (MRRE) which can be used for worst case error analyses. McKeeman [McKe67] proposed the average relative representation error (ARRE) and compared floating-point number systems with different radices. Brent [Bren73] suggested to use the root mean square (RMS) error criterion to discuss the choice of a radix for floating-point numbers of fixed length.

The roundoff methods used in conventional arithmetic are not required in the on-line algorithms, since simple truncation produces no bias [Aviz62]. We study in this paper the errors resulted from the FLPOL algorithms for addition/subtraction and multiplication in terms of the absolute error and the MRRE. The logarithmic distribution [Hamm70] is normally used as an approximation to the distribution of mantissa for conventional floating-point arithmetic. However, the statistical properties of redundant floating-point numbers have not been established, and therefore the ARRE or the RMS criterion can not be discussed at this moment.

2. RANGE OF REDUNDANT MANTISSA

Due to the nature of the digit selection function used in the on-line algorithms, the result digits are always chosen so that the computed result of an FLPOL operation is within a certain error range from the true result. This property is briefly stated in the following theorem:

Theorem 2.1: The absolute error incurred by terminating the FLPOL algorithms for addition, subtraction and multiplication at the $k+\delta$ th recursive step satisfies the inequality

$$|\epsilon_k| < [\frac{1+\Delta}{2} + 2\rho\frac{r^{-\delta}}{r-1}]r^{-k+z_e} \tag{2.1}$$

where ϵ_k is the absolute error due to truncating the computed result at the $(k+\delta)$-th step, r is the radix, ρ is the maximum digit that could appear in the operand mantissa and z_e is the exponent of the result.

Proof: Let two redundant operands be $x=x_f r^{x_e}$ and $y=y_f r^{y_e}$.

(i) Addition/Subtraction

Without loss of generality, we may assume $x_e \geq y_e$. Then the operand y is rewritten in the aligned form

$$y = r^{x_e}\sum_{j=1}^{m+1} y_j r^{-(j+i_d)}, \tag{2.2}$$

where i_d is the difference of the exponents, i.e., $0 \leq i_d = x_e - y_e < m+1$.

The absolute error committed at the $(k+\delta)$-th step is given by

$$|\epsilon_k| = |\hat{z} - Z_k| \tag{2.3}$$

where $\hat{z}$ is the true sum and Z_k denotes the computed result at the $(k+\delta)$-th recursive step. We must note that in on-line fixed-point addition the algorithm outputs δ zeros in the beginning if there is no overflow. The true sum is written in the form

$$\begin{aligned}\hat{z} &= r^{x_e}[\sum_{j=1}^{i_d} x_j r^{-j} + \sum_{j=i_d+1}^{m+1}(x_j + y_{j-i_d})r^{-j} \\ &\quad + \sum_{j=m+2}^{m+1+i_d} y_{j-i_d} r^{-j}] \\ &= r^{x_e}\sum_{j=1}^{m+1+i_d}(x_j + y_{j-i_d})r^{-j}\end{aligned} \tag{2.4}$$

where $x_j=0$ for $j > m+1$ and $y_j=0$ for $j \leq 0$.

If we stop the recursion at the $(k+\delta)$-th step, then we have obtained the partial remainder $w(k+\delta)$ and the result digit $d_{k+\delta}$, but the operand digits $x_{k+\delta+1}$ and $y_{k+\delta+1-i_d}$ have not been included in the recursion. Then from the recursion of the FLPOL addition we obtain the following relation:

$$\hat{w}(k+\delta+1) = r[w(k+\delta) - d_{k+\delta}] + r^{-\delta}(0+0)$$

$$= r^{k+1}\sum_{j=1}^{k+\delta}(x_j + y_{j-i_d})r^{-j} - r^{k+\delta+1}\sum_{j=1}^{k+\delta} d_j r^{-j}$$

* Supported in part by the ONR Contract No. N00014-79-C-0866 (Research in Distributed Processing).

That is,

$$r^{-(k+1)}\hat{w}(k+\delta+1) = \sum_{j=1}^{k+\delta}(x_j+y_{j-i_d})r^{-j} - r^{\delta}\sum_{j=1}^{k+\delta}d_j r^{-j} \tag{2.5}$$

where $y_j=0$ for $j\leqslant 0$. Multiplying both sides of (2.5) by r^{x_e}, we obtain

$$\begin{aligned} Z_k = r^{z_e}\sum_{i=1}^{k} z_i r^{-i} &= (r^{\delta}\sum_{j=1}^{k+\delta} d_j r^{-j})r^{x_e} \\ &= r^{x_e}[\sum_{j=1}^{k+\delta}(x_j+y_{j-i_d})r^{-j} \\ &\quad - r^{-(k+1)}\hat{w}(k+\delta+1)] \end{aligned} \tag{2.6}$$

where $z_e=x_e-L$ if no overflow takes place, and $z_e=x_e+1$ and $z_1=1$ otherwise. From (2.3), (2.4) and (2.6),

$$\begin{aligned} |\epsilon_k| &= r^{x_e}\Big|\sum_{j=k+\delta+1}^{m+1+i_d}(x_j+y_{j-i_d})r^{-j} \\ &\quad + r^{-(k+1)}\hat{w}(k+\delta+1)\Big| \\ &\leqslant [\hat{w}(k+\delta+1)r^{-(k+1)} + 2\rho\sum_{j=k+\delta+1}^{m+1+i_d} r^{-j}]r^{x_e} \end{aligned} \tag{2.7}$$

Since $|\hat{w}(k+\delta+1)| < r\frac{1+\Delta}{2}$ and in the worst case $i_d=0$ and $x_j=y_j=\rho$, we obtain (2.1). The righthand side of inequality (2.1) is derived if an infinite precision of the operand representation is allowed. In practice, the precision of the operands is limited and the error would not exceed that value. For subtraction the same conclusion is derived by changing the sign of the addend y. Hence the theorem is correct for addition and subtraction.

(ii) Multiplication

In case of multiplication, the problem is simpler than in addition, because the mantissa alignment is not needed. Let us assume that the mantissa of the product has L leading zeros before normalization. Then the FLPOL multiplication algorithm discards $\delta+L$ initial zeros and adjusts the exponent of the result to x_e+y_e-L.

For FLPOL multiplication, we have for the true product

$$\begin{aligned} \hat{z} &= r^{x_e+y_e}\sum_{j=1}^{m+1}x_j r^{-j}\sum_{j=1}^{m+1}y_j r^{-j} \\ &= r^{x_e+y_e}\sum_{j=1}^{m+1}(x_j Y_j + y_j X_{j-1})r^{-j} \end{aligned} \tag{2.8}$$

where $X_{j-1}=\sum_{n=1}^{j-1}x_n r^{-n}$, and $Y_j=\sum_{n=1}^{j}y_n r^{-n}$. Let us denote x_e+y_e, the exponent before normalization, as z_e. From the recursion of FLPOL algorithm, it follows that

$$\begin{aligned} Z_k = r^{z_e}\sum_{i=1}^{k} z_i r^{-i} &= r^{\delta+z_e}\sum_{j=1}^{k+\delta} d_j r^{-j} \\ &= [\sum_{j=1}^{k+\delta}(x_j Y_j + y_j X_{j-1})r^{-j} \\ &\quad - r^{-(k+1)}\hat{w}(k+\delta+1)]r^{z_e} \end{aligned} \tag{2.9}$$

Hence, the absolute error becomes

$$|\epsilon_k| = r^{z_e}\Big|r^{-(k+1)}\hat{w}(k+\delta+1) + \sum_{j=k+\delta+1}^{m+1}(x_j Y_j + y_j X_{j-1})r^{-j}\Big| \tag{2.10}$$

Since $|X_{j-1}|, |Y_j| < \frac{\rho}{r-1}$ and $|w(k+\delta+1)| \leqslant r\frac{1+\Delta}{2}$, we obtain

$$|\epsilon_k| < \left[\frac{1+\Delta}{2} + 2\rho^2\frac{r^{-\delta}}{(r-1)^2}\right]r^{-k+z_e} < \left[\frac{1+\Delta}{2} + 2\rho\frac{r^{-\delta}}{r-1}\right]r^{-k+z_e}.$$

Since $z_e=x_e+y_e$ is the exponent value before post-normalization, (2.1) is valid for FLPOL multiplication also. Hence, Theorem 2.1 is proved. □

This theorem is of basic importance, because it leads to the derivation of the redundant mantissa with the smallest magnitude and provides the proof that the results of the FLPOL operations are quasi-normalized for higher radices.

With regard to the mantissa of FLPOL operations, the following theorem is provided.

Theorem 2.2: The absolute value of the redundant mantissa is no less than $\left[\frac{1-\Delta}{2} - \frac{2\rho r^{-\delta}}{r-1}\right]r^{-1}$ if the mantissa is produced by the FLPOL algorithms. In other words,

$$|z_f| > \left[\frac{1-\Delta}{2} - \frac{2\rho r^{-\delta}}{r-1}\right]r^{-1}. \tag{2.11}$$

Proof: Suppose the result has L leading zeros and the most significant nonzero digit of the result is one. The overflow is considered to be the case in which $L=-1$. Obviously the leading zeros are removed by the FLPOL algorithms, and we have a digit '1' or '-1' at the $(\delta+L+1)$-st recursive step. From Theorem 2.1, this partially computed result has an absolute error of magnitude

$$|\epsilon_{L+1}| < \left[\frac{1+\Delta}{2} + \frac{2\rho r^{-\delta}}{r-1}\right]r^{-(L+1)+z_e}.$$

Then the minimum bound of the result is given by the inequality

$$|z| \geqslant r^{-(L+1)+z_e} - |\epsilon_{L+1}| > \left[\frac{1-\Delta}{2} - \frac{2\rho r^{-\delta}}{r-1}\right]r^{-1+z_e-L}$$

Hence (2.11) follows. □

The RHS of (2.11) is

$$z_{fmin} = \left[\frac{1-\Delta}{2} - \frac{2\rho r^{-\delta}}{r-1}\right] r^{-1} \tag{2.12}$$

For multiplication we obtain the range of the mantissa in a similar manner

$$|z_f| > \left[\frac{1-\Delta}{2} - \frac{2\rho^2 r^{-\delta}}{(r-1)^2}\right] r^{-1}. \tag{2.13}$$

$$\geqslant z_{fmin}$$

z_{fmin} is the smallest absolute value of the redundant mantissa of the result if the operands are represented with an infinite precision. For instance, in case $r=\rho-1$, $\delta=1$ and $\Delta=0$, (2.12) yields $z_{fmin}=\frac{1}{2}r^{-1}-2r^{-2}$. This value can be derived also by an inductive method.

Consider the recursion for on-line addition:

begin OLADD
$w(0)\leftarrow 0$; $z_0\leftarrow 0$;
for j=1 **step** 1 **until** $m+\delta+1$ **do**
begin
w(j) $\leftarrow$ r[w(j-1)-z_{j-1}] + $r^{-\delta}(x_j+y_j)$;
z_j $\leftarrow$ SEL[w(j)]
end
end OLADD

Let us assume $z_j=0$ for $1\leqslant j\leqslant k-1$ and $z_j=1$. Therefore, we must have

$$0.5 \leqslant w(k) < 1.5$$

Then for j=k+1 we have

$$w(k+1) = r[w(k)-1] + r^{-\delta}(x_{k+1}+y_{k+1})$$

To find the smallest absolute value of redundant mantissa, we find the minimum value of w(k+1). The minimum value is obtained when $w(k)-z_k=-(1+\Delta)/2$ and the digit inputs $x_{k+1}=y_{k+1}=-\rho$. That is,

$$\min w(k+1) = -(1+\Delta)r/2 - 2\rho r^{-\delta}.$$

Since for addition full precision comparison is possible, we have $\Delta=0$. For $\delta=1$ and $\rho=$r-1, we have

$$\min w(k+1) = -\frac{r}{2}-2+\frac{2}{r}.$$

For r>4,

$$z_{k+1} = -(\frac{r}{2}+2).$$

Then it follows that

$$\min w(k+2) = \frac{2}{r}, \text{ and } z_{k+2}=0.$$

It can be proved by induction that $z_k=0$ for all j>k+2. Since the leading zeros are removed by the FLPOL algorithms, the smallest redundant mantissa with the msd 1 becomes

$$\min |z_f| = 0.1d0000....,$$

where d=-(r/2+2) for $\rho=r-1$, $\delta=1$ and $\Delta=0$. Therefore, min $|z_f| = r^{-1}-(r/2+2)r^{-2} = \frac{1}{2}r^{-1}-2r^{-2} = z_{fmin}$. Thus the ranges of redundant mantissa obtained by the both methods agree.

Corollary: The FLPOL arithmetic operations can always yield a result with quasi-normalized mantissa for any radix higher than 2 if there is no restriction on the number of delays and the number of comparison digits.

Proof: It is sufficient to show that $|z_{fmin}|>r^{-2}$. From (2.12) it follows that

$$r^{-1} \leqslant \frac{1-\Delta}{2} - \frac{2\rho r^{-\delta}}{r-1}. \tag{2.14}$$

Let $\Delta=2r^{1-\beta}$, where β is the number of comparison digits used in the digit selection function. If there is no restriction on the values of δ and β, it is possible to satisfy (2.14) for any radix r>2. Hence $z_{fmin}\geqslant r^{-2}$ for r>2. □

For radices higher than 6, $z_{fmin}>r^{-2}$ with $\delta=1$. For lower radices except 2, it is possible to produce quasi-normalized results by using larger value of δ. For radix 2, z_{fmin} is no less than r^{-3}.

3. REPRESENTATION ERRORS

(a) The Absolute Error

The maximum bound of the absolute error is easily derived from Theorem 2.1 by setting k=m+1 in equation (2.1). However, the maximum error bound can be improved if we study the FLPOL arithmetic operations carefully. For static error analysis, we study the effect of truncating the result at (m+1)-st digit assuming the operands are free of errors.

A. Addition
(1) overflow in the sum
(i) $i_d\leqslant\delta-1$

After $m+\delta$ steps all the digits of the operands are included in the recursion , and (2.7) leads to

$$|\epsilon_m| = |r^{-(m+1)}\hat{w}(m+\delta+1)|r^{x_e}$$

$$\leqslant \frac{1+\Delta}{2} r^{-(m+1)+z_e} \tag{3.1}$$

where $z_e=x_e+1$.

(ii) $i_d>\delta-1$

Similarly from (2.7),

$$|\epsilon_m| = r^{x_e}|y_{m+\delta-i_d+1}r^{-(m+\delta+1)}+\ldots$$

$$+ y_{m+1}r^{-(m+i_d+1)} + r^{-(m+1)}\hat{w}(m+\delta+1)|$$

$$< \left[\frac{1+\Delta}{2} + \frac{\rho r^{-\delta}}{r-1}\right] r^{-(m+1)+z_e} \tag{3.2}$$

where $z_e=x_e+1$
(2) No overflow in the sum

(i) $i_d \leqslant \delta$

After $m+\delta+1$ recursive steps all the operand digits are included in the recursion, and (2.7) leads to

$$|\epsilon_{m+1}| \leqslant |r^{-(m+2)}\hat{w}(m+\delta+2)|r^{x_e}$$

$$= \frac{1+\Delta}{2} r^{-(m+1)+z_e}, \tag{3.3}$$

where $z_e=x_e$
(ii) $i_d > \delta$

Similarly from (2.7),

$$|\epsilon_{m+1}| \leqslant r^{x_e}|y_{m+\delta+2-i_d}r^{-(m+\delta+2)}+\ldots$$

$$+ y_{m+1}r^{-(m+i_d+1)}+r^{-(m+2)}\hat{w}(m+\delta+2)|$$

$$< \left[\frac{1+\Delta}{2} + \frac{\rho r^{-\delta}}{r-1}\right] r^{-(m+1)+z_e}, \tag{3.4}$$

where $z_e=x_e$.

Summarizing the above cases and considering $\Delta=0$ for addition, the absolute error e_a of FLPOL addition is bounded by the following inequality:

$$|e_a| < [\frac{1}{2} + \frac{\rho r^{-\delta}}{r-1}]r^{-(m+1)+z_e} \tag{3.5}$$

B. Multiplication

After $m+L+\delta+1$ recursive steps, all the digits of the operands are included in the recursion, since $\delta \geqslant 1$ and $L \geqslant -1$. From (2.10), the absolute error e_m of FLPOL multiplication satisfies the inequality

$$|e_m| \leqslant \frac{1+\Delta}{2} r^{-(L+m+1)+x_e+y_e}$$

$$= \frac{1+\Delta}{2} r^{-(m+1)+z_e}, \tag{3.6}$$

where $z_e=x_e+y_e-L$.

We see from (3.5) that the absolute error of FLPOL addition is nearly rounded for higher radices or a large value of on-line delay. (3.6) also shows that the absolute error of FLPOL multiplication is nearly rounded if the number of comparison digits is large enough. These results are quite natural because the digit selection function is a rounding procedure.

(b) The Maximum Relative Representation Error

The maximum relative representation error (MRRE) for redundant floating-point numbers is defined as

$$MRRE = \frac{\max|\hat{z}_f - z_f|}{\min|\hat{z}_f|}, \tag{3.7}$$

where $\hat{z}=r^{z_e}\hat{z}_f$ is the true result of an arithmetic operation and $z=r^{z_e}z_f$ is the computed value.

(i) Addition

From (2.12) and (3.5) we obtain the expression for the MRRE of FLPOL addition as

$$E_a = \frac{\frac{1}{2} + \frac{\rho r^{-\delta}}{r-1}}{\frac{1}{2} - \frac{2\rho r^{-\delta}}{r-1}} r^{-m} \tag{3.8}$$

(ii) Multiplication

Similarly from (2.13) and (3.6), it follows that

$$E_m = \frac{\frac{1+\Delta}{2}}{\frac{1-\Delta}{2} - \frac{2\rho^2 r^{-\delta}}{(r-1)^2}} r^{-m} \tag{3.9}$$

The values of the MRRE for addition and multiplication are shown in Table 3.1 for different combination of r, ρ and δ.

The MRRE of redundant floating-point numbers can be used to compare the characteristics of different floating-point number systems [McKe67], and also it may be used for worst case error analyses [Wilk63] of the computations using on-line arithmetic. As we see in the table, a larger radix with smaller ρ yields better figures of the MRRE.

			MRRE (xr^{-m})	
r	ρ	δ	addition ($\Delta=0$)	multiplication ($\Delta=2r^{-2}$)
8	4	2	1.0555556	1.0874243
	5	1	1.8333333	1.4450402
	6	1	2.125	1.7147402
	7	1	2.5	2.2
10	5	2	1.0340909	1.0540954
	6	1	1.5454545	1.2714681
	7	1	1.6774193	1.3820676
	9	1	2.0	1.7586207
16	8	2	1.0126050	1.0203185
	9	1	1.2647059	1.1170765
	10	1	1.3	1.1438424
	15	1	1.5	1.3578947
32	16	2	1.0030303	1.0049617
	17	1	1.1103896	1.0432058
	18	1	1.1173913	1.0481742
	31	1	1.2142857	1.1476510

Table 3.1: The Maximum Relative Representation Error

4. ERROR GROWTH IN ADDITION

A result of simulation is shown in Fig. 4.1. A constant is added repeatedly, and the error growth is measured for both on-line

and conventional arithmetic. In conventional rounding arithmetic, the MRRE is given by $\frac{1}{2}r^{1-m}$. For instance, in radix-10 8-digit number representation system the MRRE is $5.0x10^{-8}$. In FLPOL arithmetic, one additional digit is computed to prevent loss of significant digit in the result mantissa because of quasi-normalization. As a result, the MRRE of FLPOL arithmetic is maintained smaller than that of conventional arithmetic. In the radix-10 9-digit maximally redundant representation, the MRRE for FLPOL addition is $2.0x10^{-8}$ as shown in Table 3.1. As expected from the comparison of the MRRE's, we observe in Fig. 4.1 that the error growth of FLPOL addition is smaller than that of conventional addition. The relative error of FLPOL addition grows relatively rapidly near the points where the number of repetition is approximately 130. This is due to quasi-normalization of the result which causes loss of one significant digit.

The worst case absolute error of

$$Y(N) = [Y(N-1) + X](1+E_a) \qquad (4.1)$$

is approximately given by $\frac{1}{2}E_aN(N-1)X$ neglecting higher order terms. Since the true value of Y(N) is NX, the relative error is $e_r=\frac{1}{2}E_a(N-1)$. If we substitute the MRRE of FLPOL addition $E_a=2.0x10^{-8}$ and the number of iterations N=201, we obtain $e_r=2x10^{-6}$. In Fig. 4.1, the relative error of on-line arithmetic at N=200 is approximately one-seventh of the worst case error.

5. CONCLUSION

The static error characteristics of the FLPOL arithmetic operations have been discussed in detail. General properties of the redundant mantissa have been studied. By using a method of error analysis, the range of the redundant mantissa has been derived for general redundant representation. The range can be also derived by an inductive method. As a result, the FLPOL operations are proved to yield quasi-normalized results. The MRRE has been defined for redundant floating-point numbers using the absolute error and the range of the redundant mantissa. The MRRE is improved if a higher radix with smaller ρ is used. The MRRE of FLPOL operations can be used to compare the characteristics of different floating-point number systems and can be used for the worst case error analysis of computations using FLPOL arithmetic also.

6. REFERENCE

[Aviz61] Avizienis, A., "Signed-digit number representation for fast parallel arithmetic," IRE Trans. Electron. Comput., vol. EC-10, no. 3, pp. 389-400, 1961.

[Aviz62] Avizienis, A., "A flexible implementation of digital computer arithmetic," IFIP proceedings, pp. 664-670, 1962.

[Bren73] Brent, R. P., "On the precision attainable with various floating-point number systems," IEEE Trans. Comput., vol. C-22, no. 6, pp. 601-607, June 1973.

[Erce77] Ercegovac, M. D., "A general hardware-oriented method for evaluation of functions and computations in a digital computer," IEEE Trans. Comput., vol. C-26, no. 7, pp. 667-680, July 1977.

[Hamm70] Hamming, R. W., "On the distribution of numbers," Bell Syst. Tech. J., vol. 49, no. 8, pp. 1609-1625, Oct. 1970.

[Hwan79] Hwang, K., "Computer arithmetic," Chapter 10, J. Wiley & Sons, N.Y., 1979.

[Irwi77] Irwin, M. J., "An arithmetic unit for on-line computation," Ph.D. dissertation, Dept. Computer Science, Univ. of Illinois at Urbana-Champaign, Tech. Rept. 873, May 1977.

[McKe67] McKeeman, W. M., "Representation error for real numbers in binary computer arithmetic," IEEE Trans. Electron. Comput., vol. EC-16, no. 5, Oct. 1967.

[TrEr77] Trivedi, K. S. and Ercegovac, M. D., "On-line algorithms for division and multiplication," IEEE Trans. Comput., vol. C-26, no. 7, pp. 681-687, July 1977.

[Wilk63] Wilkinson, J. H., "Rounding errors in algebraic processes," Prentice Hall, 1963.

[WaEr81] Watanuki, O. and Ercegovac, M. D., "Floating-point on-line arithmetic: Algorithms," Proceedings of 5th Symposium on Computer Arithmetic, 1981.

[Yohe73] Yohe, J. M., "Roundings in floating-point arithmetic," IEEE Trans. Comput., vol. C-22, no. 6, pp. 577-586, June 1973.

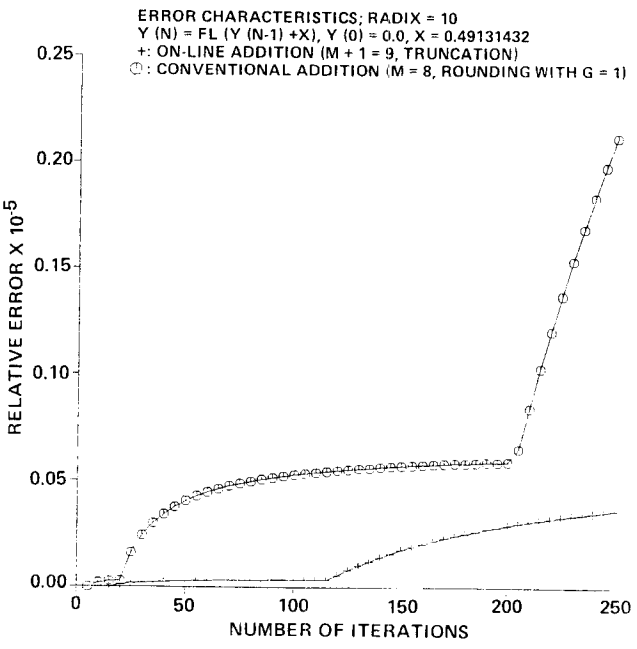

Figure 4.1: Error Growth in Addition: A Comparison