

AD-A113 961

BEDFORD RESEARCH ASSOCIATES MA

F/S 12/1

IMPACT OF FINITE PRECISION ARITHMETIC ON ALGORITHM DESIGN - PAR--ETC(U)

AUG 81 P TSIPOURAS, C STEELE

F19628-80-C-0124

UNCLASSIFIED

SR-3

AF6L-TR-81-0216

NL

1 of 1
AD A081



END
DATE
FILMED
DTIC

12

AFGL-TR-81-0216

IMPACT OF FINITE PRECISION ARITHMETIC ON
ALGORITHM DESIGN - PART I.

P. Tsipouras
C. Steele

Bedford Research Associates
2 DeAngelo Drive
Bedford, Massachusetts 01730

AD A 1 1 3 9 8 1

Scientific Report No. 3

3 August 1981

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

DTIC
SELECTED
APR 27 1982
S H D

DTIC FILE COPY

82 04 27 133

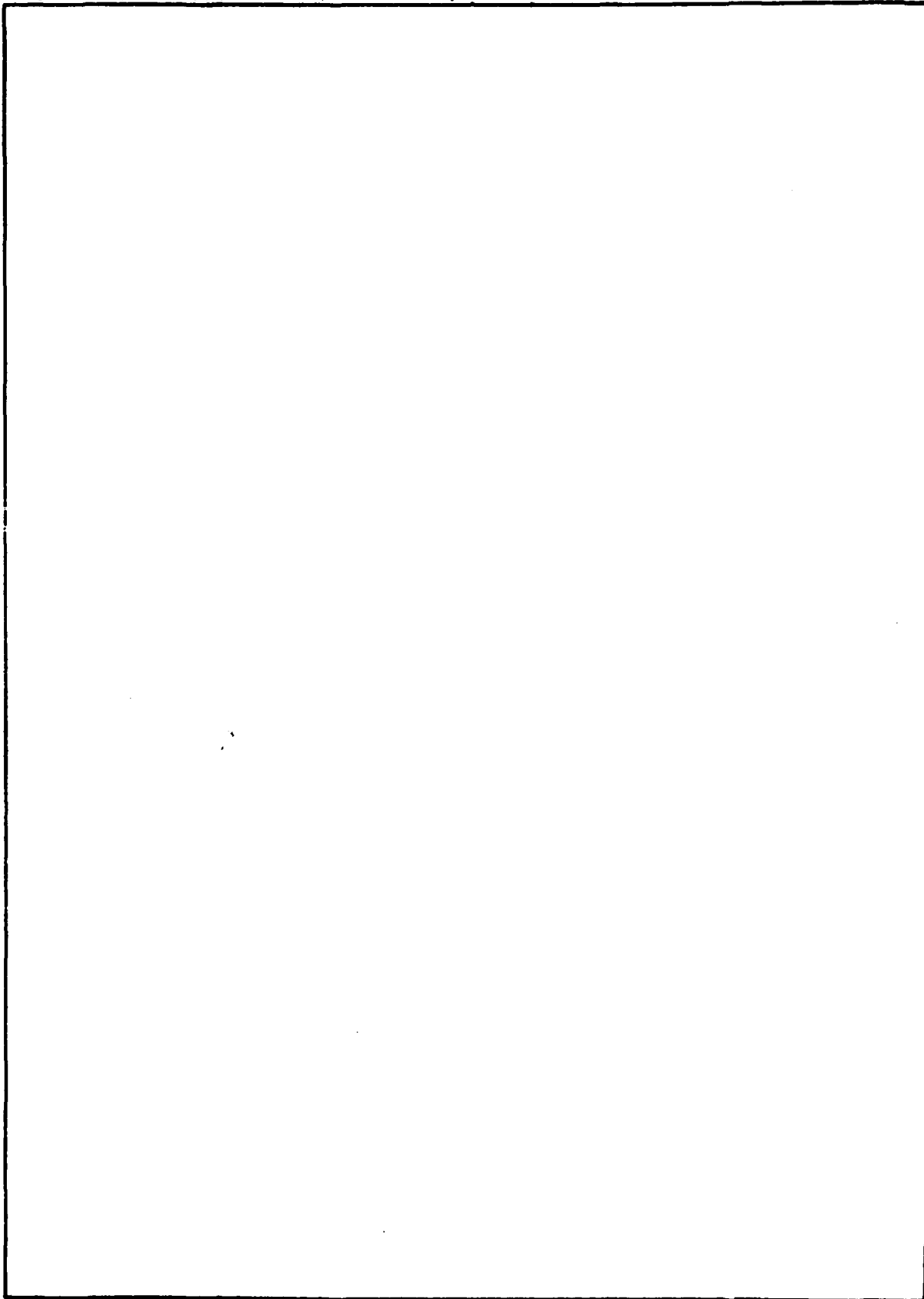
Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFGL-TR-81-0216	2. GOVT ACCESSION NO. AD-A113 981	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Impact of Finite Precision Arithmetic on Algorithm Design - Part 1		5. TYPE OF REPORT & PERIOD COVERED Scientific Report No. 3
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) P. Tsipouras * C. Steele		8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bedford Research Associates 2 DeAngelo Drive Bedford, MA. 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 9993XXXX
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratories Hanscom AFB, MA. 01731 Monitor/Paul Tsipouras/SUNA		12. REPORT DATE 3 August 1981
		13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES * Air Force Geophysics Laboratories Analysis and Simulation Branch (SUWA) Hanscom AFB, MA. 01731		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Finite precision arithmetic, algorithms, mathematical software, digital computer.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A brief discussion of the impact of finite precision arithmetic on computer algorithms and mathematical software.		

**DTIC
SELECTED**
 APR 27 1982
H

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

In the following pages we wish to discuss briefly floating point computation as performed on a typical digital computer. Our objective will be two-fold: to illustrate the peculiarities of arithmetic in such an environment caused by the imprecise representation of the real number system, and to indicate how various choices in representation and arithmetic algorithms impinge on mathematical software.

1. Basic Issues

As usual, we assume a positional or polynomial representation of whole numbers, i.e. for a given value x we have

$$x = P(\beta) = \sum_{i=0}^{n-1} d_i \beta^i \quad (1)$$

where P is the base or radix of the representation of x , n the number of base β digits d_0, d_1, \dots, d_{n-1} with $d_i \in \{0, 1, \dots, \beta-1\}$ often express the base β representation of x in (1) as

$$x = (d_{n-1} d_{n-2} \dots d_2 d_1 d_0)_{\beta} \quad (2)$$

As usual a sign is prefixed to (1) to extend the representation to the integers. Note that for fixed n , it is obvious that (1) or (2) can represent only values in the range

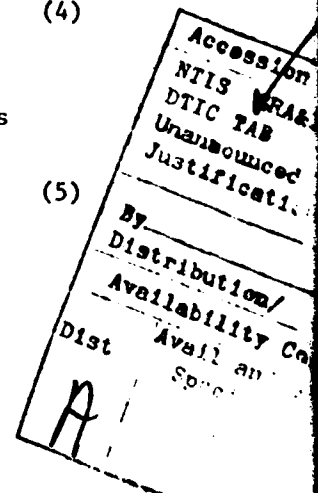
$$0 \leq x \leq \beta^n - 1 \quad (3)$$

(or, with a sign, $-\beta^n + 1 \leq x \leq \beta^n - 1$). To extend (1) for range of the rationals, negative indices or exponents are permitted:

$$x = P_{n,m}(\beta) = \sum_{i=-m}^n d_i \beta^i \quad (4)$$

yielding $n+m$ digit base β rationals, i.e., we can rewrite (4) as

$$x = (d_{n-1} d_{n-2} \dots d_2 d_1 d_0 \cdot d_{-1} d_{-2} \dots d_{-m})_{\beta} \quad (5)$$



Example 1.

$$P_{4,4}(10) = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1} + 6 \cdot 10^{-2} + 7 \cdot 10^{-3} + 8 \cdot 10^{-4} = (1234.5678)_{10}$$

$$P_{3,3}(2) = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = (110.111)_2 = (6.875)_{10}$$

Note that in (4) or (5) if the number of base β digit to the right of the radix point is fixed a priori regardless of the number we wish to represent we have a fixed point notation or number system which is limited to representing magnitudes in the range

$$0 \leq x \leq \beta^n - 1$$

as before, but now with the smallest non-zero magnitude representable being β^{-m} . With the addition of a scale factor, s range of representable numbers can be expressed as

$$\beta^s (\beta^n - 1) \leq x \leq \beta^s (\beta^n - 1)$$

The range of useful values in our system of representation can be greatly increased if we use a scheme similar to "scientific notation" for physical constants, i.e. quantities are represented in terms of signed fraction or mantissa and a signed exponent or characteristic. Thus we have now to specify the number of digits, e and the base, β_e , of the exponent and the number of digits, m and the base, β_m of the mantissa. Thus we have a two part representation (f,c) where f , the fraction, is a fixed point number with base β_m , m β_m - digits and scale factor s and c , the exponent a base β_e , e digit fixed point number with scale factor s_e (usually $s_e = 0$ hence the exponent is an integer). In addition, the exponent may be biased, i.e. if the exponent lies in the range $-M_1 \leq \text{exponent} \leq M_2$ then to avoid the explicit sign, M_1 may be added to all exponents;

hence an excess $-M_1$ notation. If the scale factor, s is zero, then the radix point is at the right of the mantissa (e.g. Burroughs, CDC hardware) while a factor of $S = -m$ yields an implied point at the left hand side (e.g. the IBM S/360/ S/370) or $|f| < 1$, hence $\beta_m^m f$ is an integer and $-\beta_m^m \leq f < \beta_m^m$.

Note that up to this point we have distinguished between β_e , the exponent base and β_m , the mantissa base. Almost without exception these are powers of the hardware base, β_v which is usually 2. (e.g. for the IBM S/360-S/370, $\beta_e = 2$, $\beta_m = 16$). To avoid fractional exponents, we refer to β , the floating point base/radix and assume $\beta = \beta_m$. Clearly the largest e digit exponent then is $\beta_e^e - 1$. Hence the largest representable number is equal to

$$(\text{the largest mantissa}) \times \beta^{\beta_e^e - 1} \quad (7)$$

Assuming $\beta = \beta_v^k$ then the right hand terms becomes $\beta_v^k (\beta_e^e - 1)$; we define the exponent range to be $k (\beta_e^e - 1)$. With respect to the mantissa in (7) if the scale factor, s , is zero, then the largest mantissa is $\beta^m - 1$, the smallest, 1. If $s = -m$ on the other hand, then the largest mantissa is

$$\beta^{-m} (\beta^m - 1) - 1 - \beta^{-m} \approx 1. \quad (8)$$

and the smallest is β^{-m} .

Thus, assuming an exponent scale factor of 0, we can denote our floating point number system by FL (β_e, β, m, e, s) (9) where β_e is the base of the exponent, β the base (of the mantissa), m , the number of digits in the mantissa, e the number of digits in the exponent and s the (mantissa) scale factor.

If $s = -m$, then we can simplify (9) to FL (β_e, β, m, e) (10) with (10) the representable values of the system, denoted $s (\beta_e, \beta, m, e)$, are given by

$$\beta^{1-m-\beta_e^e} \leq x \leq (1 - \beta^{-m}) \beta^{\beta_e^e - 1} \quad x \neq 0 \quad (11)$$

$$-(1 - \beta^{-m}) \beta^{\beta_e^e - 1} \leq x \leq -\beta^{1-m-\beta_e^e}$$

If $\beta = \beta_e = 2$, then $S(2,2,m,e)$ becomes

$$2^{1-m-2^e} \leq x \leq (1-2^{-m}) 2^{2^e-1} \quad x \neq 0 \quad 2^{2^e-1} \quad (12)$$

$$-(1-2^{-f}) 2^{2^e-1} \leq x \leq -2^{1-m-2^e}$$

The precision of the set $S(\beta_e, \beta, m, e)$ is defined to be the number of digits representable in the mantissa, normally in base β digits, i.e. $\{0, 1, \dots, \beta-1\}$; for purposes of comparison we speak of the binary precision, the number of the bits (base 2 digits) representable in the mantissa. The range of FL $(\beta_e, \beta, m, e, s)$ is defined to be the largest representable magnitude, hence

$$\beta^s (\beta^m - 1) \beta^{\beta^e - 1} \quad (13)$$

or with $\beta=2$ and fractional mantissae, range FL $(2,2,f,e) = (1-2^{-m}) 2^{2^e-1}$.

Example 2.

For some typical hardware families we have

<u>Machine</u>	<u>Word Size</u> <u>(bits)</u>	<u>β</u>	<u>Exponent</u> <u>(bits)</u>	<u>Mantissa</u> <u>(bits)</u>	
Burroughs 6700/7700	47 plus tag	8	7 (S-M)	39 (integer)	S-M
CDC 6600/ Cyber 70	60	2	11 (Excess 2^{10})	48 (integer)	1's C
DEC PDP-11	32	2	8 (Excess 2^7)	23 (fraction)	S-M
Honeywell H8200	48	2	7 (Excess 2^6)	40 (Binary)	S-M
		10		10 (BCD) (fraction)	
IBM S/370	32	16	7 (Excess 2^6)	24 (fraction)	S-M

Thus far, we have ignored the questions of uniqueness; clearly for a given real value x we wish to have a representative within $S(\beta_e, \beta, m, e, s)$ which best approximates x in some sense. However, since $m\beta^e = \beta^{-a} m \beta^{a+e}$ for any integer, a , we choose a normalized representative, i.e. one such that the most significant digit of the mantissa is non-zero, with the exponent adjusted accordingly by a suitable choice of a . The corresponding Normalized Floating Point Number System is denoted NFL $(\beta_e, \beta, m, e, s)$, or if fractional mantissae are assumed NFL (β_e, β, m, e) ($\therefore s = -m$). Thus the smallest representable (non-zero) magnitudes are respectively

$$\beta^{m-1} (s=0) \text{ and } \beta^{-m} (\beta^{m-1}) = \beta^{-1}$$

Example 3.

For a binary system ($\beta = \beta_e = 2$) with m bits for the mantissae, e for the exponent, assuming $s = -m$, then if we denote the set of values in NFL $(2, 2, m, e)$ by NS $(2, 2, m, e)$, $x \in \text{NS}(2, 2, m, e)$ if

$$2^{-2e} \leq x \leq (1-2^{-m}) 2^{2^e-1}, \quad x = \pm 0,$$

or $(1-2^{-m}) 2^{2^e-1} \leq x \leq -2^{-2e}$

2. Relations In The Parameters

In the preceding sections we discussed the basic normalized floating point representation, with the tacit assumption that the number of base β digits available for the mantissa and the number of exponent (base β_e) digits are determined by hardware considerations. Here we wish to examine the choices available or implied for these parameters and the tradeoffs between allocations of resources to one versus another of them. First, let us assume the mantissa size, m , is fixed, and that exponents are positive. Then the representation ratio of the number of values using a higher base that can be represented to the number of binary values representable in the range of binary numbers, representable with the available number of bits.

For example, for $\beta=2$, in a normalized floating point system only half of the possible representable values are used, for a given mantissa size, i.e. those with the most significant bit = 1. On the other hand, when a leading 0 is permitted ($\beta = 4$), 50% more values are representable. Similar increases occur for higher values of β .

Now consider NFL (β_e, β, m, e) ($\therefore s = -m$); there are 2^e different exponents and 2^{m-1} different normalized mantissae representable, if $\beta_e = \beta = 2$. Thus the total number of positive representable values with positive exponents is 2^{e+m-1} ; since the largest representable mantissa is = 1 and the largest representable exponent is $2^e - 1$, the largest representable binary number is $= 2^{2^e - 1}$. If we contrast this with the case $\beta=2^k$, with numbers of the form $m \cdot \beta^e$ and estimate the number of values less than the largest representable binary number ($2^{2^e - 1}$), assume $M \gg 1$ and choose p such that $\beta^p \approx 2^{2^e - 1}$ so that $p \log_2 \beta \approx 2^e - 1$. Thus, the number of representable values between β^{-1} and β^p is approximately

$$(2^{m-1} + 2^{m-2} + \dots + 2^{m-\log \beta}) (p+1) = 2^m (1-\beta^{-1}) (p+1)$$

where m is the number of binary digits, not base β digits. To compute the representation ratio, we compare the total number of base β values in the range $[2^{-1}, 2^{2^e - 1}]$ to the total number of binary values in that range, i.e.

$$\begin{aligned} \text{Representation Ratio} &= \frac{\text{Base } \beta \text{ values in } [2^{-1}, 2^{2^e - 1}]}{\text{Binary values in } [2^{-1}, 2^{2^e - 1}]} \\ &\approx \frac{2^m (1-\beta^{-1}) (p+1)}{2^{m-1} 2^e} = \frac{2 (1-\beta^{-1}) (p+1)}{1 + p \log \beta} \end{aligned} \quad (14)$$

Example 4.

$$\text{Let } \beta = 16, e = 8; \text{ then } p = \frac{2^e - 1}{\log_2 \beta} = \frac{2^8 - 1}{\log_2 64} = \frac{255}{6} = 42.5$$

$$\text{Representation Ratio} \approx \frac{2(1-\beta^{-1})(p+1)}{1 - \beta^{-1} p \log \beta} = \frac{2(1-16^{-1})(43)}{1 - 16^{-1} \cdot 42.5 \cdot \log_2 16} \approx 0.475$$

It is easy to see that for fixed e, m there are about 1.875 times as many base β values representable as base 2 values. Hence, for positive exponents, $\frac{1}{4}$ (0.475/1.875) of the hexadecimal values are in the range of the binary values, and $\frac{3}{4}$ outside, for fixed e, m . Note that as we have seen, more numbers are representable over a wider range by using $\beta = 2^k, k > 1$, on the interval where the base β and binary values overlap, the binary numbers are much more densely distributed, hence do a better job at representing that interval of the real numbers.

A closely related issue to consider is the choice of β and the size of e : assume the word size $n = m + e$ is fixed, and consider the tradeoff between a choice for e (or m) and the choice of β . Obviously one criterion is the effect on the range of representable numbers. Let the exponent range be k ($\beta^e - 1$) for $\beta = 2^k$ and consider the worst-case analysis of the accuracy of representation. Clearly if $\beta = 2^k$, accuracy will decrease automatically with k , since $m - k + 1$ bits are used in the worst case.

Let \hat{x} denote the floating point representation of x in $FL(\beta_e, \beta, m, e, s)$. Then, the absolute representation error is given by $|\hat{x} - x|$, the relative representation error, $\delta(x)$, by

$$\delta(x) = \frac{|\hat{x} - x|}{x} \quad (15)$$

hence $\hat{x} = x(1 + \delta(x)) \quad (16)$

An upperbound on $\delta(x)$ is given by the smallest ϵ such that $|\delta(x)| \leq \epsilon \forall x \in FL(\beta_e, \beta, m, e, s)$; we refer to ϵ as the Maximum Relative Representation Error (MRRE) for the machine representation of x .

With β_v the hardware radix, it can be shown (Brown, Richman) that

$$\text{MMRE} = [1 - 1/\beta_v^{k-m}]^{-1} \quad (17)$$

Now, assume $m \geq k$ so that at least 1 digit of accuracy is available with $\beta = \beta_v^k$ and consider the two normalized floating point systems

$$\text{NFL}_1 = \text{NFL}(\beta_v, \beta_v^k, m, n-m), \text{NFL}_2 = \text{NFL}(\beta_v, \beta_v, m-k+1, n-(m-k+1))$$

and, since $\beta_v^{k-m} = \beta_v^{1-(m-k+1)}$, both NFL_1 and NFL_2 have the same MRRE. With respect to exponent range, since $\beta_e = \beta_v$ for both,

$$\frac{\text{Exponent Range (NFL}_2)}{\text{Exponent Range (NFL}_1)} = \frac{\beta_v^{(n-m+k-1)-1}}{K(\beta_v^{n-m}-1)} = \frac{\beta_v^{k-1}(\beta_v^{n-m}-\beta_v^{1-k})}{K(\beta_v^{n-m}-1)}$$

$$\text{and, since } \beta_v^{1-k} \leq k, \quad \frac{\text{Exponent Range (NFL}_2)}{\text{Exponent Range (NFL}_1)} > \frac{\beta_v^{k-1}}{K} \geq 1$$

If $k = \beta_v = 2$, then the exponent ranges are almost equal, but otherwise, for a fixed MRRE, the choice $\beta = 2$ clearly yields a large range of exponents, indicating that for these criteria, we should choose $\beta = 2$.

With respect to numerical computations, additional parameter relations must hold. Clearly, for $\text{NFL}(\beta_e, \beta, m, e)$, the maximum relative spacing $\epsilon = \beta^{1-m}$ is critical. For simplicity (18) below, let us denote the smallest positive number in our system by

$$\sigma = \beta^{-1} \quad (= \beta^{e_{\min}-1}, s = -m) \quad (19)$$

and the largest representable number by

$$\lambda = (1-\beta^{-m}) \beta^{\beta_e-1} \quad (= \beta^{e_{\max}} (1-\beta^{-m}), s = -m) \quad (20)$$

Clearly we assume $\beta \geq 2$, $e_{\min} \leq e_{\max}$ above. To ensure that 1 is included we require $e_{\min} \leq 1 \leq e_{\max}$ and to ensure that $\epsilon < 1$, we must have $m \geq 2$. These minimal requirements guarantee a meaningful floating point system, but to produce numerical software with proveable mathematical properties, we need a system that is both reasonably large and well balanced.

Specifically to provide a useable range for any given precision, we require that

$$\sigma, \leq \epsilon^2 \quad (21)$$

and
$$\lambda > \epsilon^{-2} \quad (22)$$

(For example, in Brown's algorithm, for the mean of a vector, we need $\sigma < \epsilon^4 \lambda$ to avoid overflow when accumulating scaled small components; in Lawson's algorithm for the Euclidean norm of a vector, we must have $\lambda > \epsilon^{-3/2}$ to avoid overflow when summing the squares of small components and $\sigma < \epsilon^2$ to avoid underflow. Thus (21) is essential for Lawson's algorithm, while (22) provides a safety factor). Assuming the usefulness of (21), (22) their realism needs to be considered. Clearly for (21), (22) to fail we would have a small range and relatively high precision. For convenience, we restate (21), (22) as

$$e \text{ min} \leq 2-2m \quad (23)$$

$$e \text{ max} \geq 2m-1 \quad (24)$$

Example 5.

An extreme example of high precision ($\beta=2$) with only 8 bits allocated to the signed exponent is provided by the DEC PDP-10 and Honeywell 6000 series; in double precision 64 bits are allocated to the (signed) mantissae from a word size of 72 bits. These machines satisfy (23), (24) by a very small margin. Assuming word size $n=72$ with $e=8$, $m=64$ we can use an implicit normalization to yield a precision of 64. Likely exponent ranges (setting one value aside for zero) would be $[-127, 127]$ or $[-126, 128]$. The proposed inequalities then are the tightest that would allow both of these possibilities. In addition to (21), (22) we would prefer

$$\sigma \lambda \approx 1 \quad (25)$$

(cf. Reinsch) but (25) is neither essential nor realistic; instead we require the weaker

$$\sigma \epsilon^{-1} < (\sigma \lambda)^{-1} < \epsilon \lambda \quad (26)$$

which may be written as $\sigma^{-2} \lambda < \epsilon$, (27)

and $\sigma \lambda^2 < \epsilon^{-1}$ (28)

(Note that (21), (22) imply $\sigma \epsilon^{-2} < \sigma \lambda < \epsilon^2 \lambda$). In Lawson's algorithm previously cited, we must have $\sigma^2 \lambda < \epsilon^{1/2}$ and $\sigma \lambda^2 \geq 1$ to ensure that the scale factors are within range, hence (27) provides a modest safety factor and (28) a larger one. Once (27) is accepted, symmetry suggests (28). Furthermore, (27) implies (28) in practice, since $\sigma \lambda \geq 1$ if the implicit radix point is at the left while $\sigma \lambda > 1$ otherwise).

(27), (28) may be re-written as

$$2 e_{\min} + e_{\max} \leq 3-m \quad (29)$$

$$e_{\min} + 2 e_{\max} \geq m+1 \quad (30)$$

If $e_{\min} + e_{\max} = 0$, then (29), (30) follow from (23), (24). However, if $e_{\min} + e_{\max} = 2m$ (e.g. with radix point on the right), then by (24), $e_{\min} \leq 3-3m$, hence $e_{\max} = 2^m - e_{\min} \geq 5m-3$ and $e_{\max} - e_{\min} \geq 8m-6$, exactly twice the exponent range specified by (23), (24). However, most existing hardware with one implied radix point on the right has relatively large exponent range and there appears to be no floating point system with $\sigma \epsilon^{-1} < \epsilon \lambda$ failing to satisfy (26).