Model Driven Analysis of Faulty IEEE-754 Scalars

J. Elliott[a,b,*], M. Hoemmen[b], F. Mueller[a]

[a] *North Carolina State University, Dept. of Computer Science, Raleigh, N.C.*
[b] *Center for Computing Research, Sandia National Laboratories, Albuquerque, N.M.*

Abstract

Incorrect computer hardware behavior may corrupt intermediate computations in numerical algorithms, possibly resulting in incorrect answers. Prior work models misbehaving hardware by randomly flipping bits in memory. We start by accepting this premise, and present an analytic model for the error introduced by a bit flip in an IEEE 754 floating-point number. We then relate this finding to the linear algebra concepts of normalization and matrix equilibration. In particular, we present a case study illustrating that normalizing both vector inputs of a dot product minimizes the probability of a single bit flip causing a large error in the dot product's result. Furthermore, the absolute error is either less than one or very large, which allows detection of large errors. Then, we apply this to the GMRES iterative solver. We count all possible errors that can be introduced through faults in arithmetic in the computationally intensive orthogonalization phase of GMRES, and show that when the matrix is equilibrated, the absolute error is bounded above by one.

Keywords: Algorithm-based fault tolerance, resilient algorithms, numerical methods

A trend in algorithm-based fault tolerance work has been to propose an algorithmic strategy, and then inject synthetic bit flips into data operated on and present either the resulting runtime or whether the final result was correct or incorrect. Examples of this methodology include a large number of related algorithm-based fault tolerance works [3–7, 11–15]. The approaches proposed are not the motivation for our work, but the fault model used to motivate and assess such works is. A common theme in algorithm-based fault tolerance is the detection and correction of errors. This is a logical approach, if an algorithm has a mechanism for knowing if its state is corrupted, then it can do something to remedy the problem. We refer to this as fine-grained detection, as the approaches attempt to recover corrupt scalars. For example, Huang and Abraham [11] proposes a scheme for detecting corruption in two dense matrices that are being multiplied together or corruption in the resulting solution matrix. To evaluate the technique, random bits are flipped in some entries of the the matrix operands or solution matrix. This approach is motivated by soft errors in the early years of scientific computing, and the detection/correction technique presented is still actively researched as a means to guard various linear algebra operations. Davies and Chen [6] propose an approach that attempts to detect and correct soft errors (bit flips) in the LU factorization of a matrix. Wu and Chen [15] present an approach for detecting and correction soft errors in Cholesky, QR, and LU factorization. Shantharam et al. [14] propose a detection/correction scheme for a sequential conjugate gradient solver.

In all works listed, the goal is to identify if an operand or result have been silently corrupted by a bit (or bits) being flipped. In the cases of [3, 11–14] the corruption is introduced into a matrix, which is then multiplied with another matrix [11] or vector [3, 12–14]. At the scalar level, this can be modeled as a faulty operand to multiplication and the corrupted result is then accumulated in a summation.

Unfortunately, few researchers have considered the characteristics of the errors that random bit flip injection introduces. This is troubling, because, as we show, the errors introduced can vary drastically. For example, Bronevetsky and de Supinski [4] injected bit flips into random locations of various numerical kernels used in scientific computing and found that for *all* kernels tested they tended to abort roughly 10-12% of

*Corresponding author
Email address: jjellio3@ncsu.edu (J. Elliott)

Table 1: Terminology

a, b	IEEE-754 scalars.
α	Exponent of scalar. The desired exponent value, e.g., 2^{-7} or 2^5.
β	Raw mantissa of scalar.
$1.\beta$	Complete mantissa $(1 + \beta)$.
$1.\xi$	Error from a bit flip in the complete mantissa $(1 + \xi)$.
$\tilde{a}$	Scalar that experiences a fault.
e	Biased exponent, the unsigned integer value stored in the exponent bits.
$bias$	Bias used in exponent storage.
η	Error introduced from a corrupt biased exponent.
η^+	η values with a positive sign.
η^-	η values with a negative sign.
N	Number of mantissa bits.
M	Number of mantissa values 2^N.
ϵ	2^{-N}.
Z	Number of exponent bits.
K	Number of biased exponents.

the time. We will show that given random bit flip injection, the expected value of the relative error is large approximately 10% of the time. Furthermore, we will show statistics that show how the expected absolute and relative errors behave given random bit flip injection into IEEE-754 scalars. We extend our models to multiplication and show that the expected error (both absolute and relative) can be forced to behave predictably using the standard numerical operations of normalization and matrix equilibration.

The point is that bit flips introduced into the representation of floating point data will behave in two broad categories: The expected error is either small (less than one) or it can extremely large. We show this in three ways: 1) we observe this effect experimentally by injecting bit flips into dot products and tallying the errors that fall into the two categories; 2) we expose this effect by using numerical analysis to bound errors arising from a bit flip in different components of the IEEE-754 representation; 3) we derive this effect by modeling the expected absolute and relative error using the analytic model of IEEE-754 scalars.

The ways errors are distributed is important when it comes to assessing ABFT techniques, because detectors may be incapable of detecting relatively small errors. The overhead introduced can also depend on the properties of the fault. For example, in [8] it is shown that large errors tend to cause more iterations than small errors. Through statistical analysis, we show that errors will be relatively small most of the time and large infrequently. This broad range of possible errors means that random injection is introducing small errors most of the time.

We also show that the distribution of errors can be skewed by standard numerical techniques such as normalization and matrix equilibration. Based on the high variability of the absolute and relative error models, we argue that researchers should instead introduce errors that are known to be detectable, while clearly indicating which errors are undetectable. It is naïve to assume by default, that an error detector will detect all possible errors. Instead, we advocate a research methodology that shows overheads given detectable errors, and also shows the overhead required to obtain a solution with a desired accuracy when an undetectable error is introduced.

1. Errors in IEEE-754 Representation

Consider a scalar represented using the IEEE-754 specification,

$$a = (-1)^{sign} \left(1 + \sum_{i=0}^{N-1} b_i 2^{i-N}\right) \times 2^{e-bias}. \tag{1}$$

Table 2: Common IEEE-754 Implementations

Common Name	Spec. Name	Mantissa Bits (N)	Exponent Bits (Z)
Half Prec.	binary16	10	5
Single Prec.	binary32	23	8
Double Prec.	binary64	52	11
Quad Prec.	binary128	112	15

We will analyze scalars represented using Eq. (1) by decomposing them into an unsigned form

$$a = \alpha \times 1.\beta, \tag{2}$$

where $\alpha = 2^{e-bias}$ and $\beta = \sum_{i=0}^{N-1} b_i 2^{i-N}$. The notation $1.\beta$ is shorthand for $1 + \beta$. The specification depends on two parameters: The number of mantissa bits N, and the number of exponent bits Z. Table 1 summarizes our notation and terminology. Common implementations of the IEEE-754 specification are listed in Table 2 as well as the parameters that can be used to generate our models. While we list various implementations, this work uses *binary64* for examples and experiments.

This analysis is different from floating point rounding analysis, e.g., Higham [10]. The most notable difference is that values can be changed drastically and bit flips do not carry. A question this work does not address is: Which bit flips (and under what conditions) would be masked by rounding effects. However, like rounding analysis the relative and absolute errors are used to measure the error introduced.

The mantissa value, $1.\beta$ is bounded above by

$$\begin{aligned} 1.\beta &= 1 + \sum_{i=1}^{N} 2^{-i} \\ &= 1 + \sum_{i=0}^{N-1} ar^i; \text{ for } a = r = 1/2 \\ &= 2 - 2^{-N} \\ &= 2 - \epsilon. \end{aligned}$$

The smallest representable value larger than 1.0 is $1 + 2^{-N}$, and 2^{-N} is often referred to as ϵ (or machine epsilon). The smallest mantissa value is obtained by letting all bits be zero, yielding $1.\beta = 1.0$. The range of the mantissa values is

$$1.0 \leq 1.\beta < 2.0. \tag{3}$$

The exponent value is not stored as a signed integer. Rather, it uses an offset, called the bias, allowing the exponent bits to represent signed and unsigned values by subtracting a bias from the stored *biased exponent*. For binary64 values, there are 11 exponent bits allowing unsigned integers in the range of $[0, 2047]$ to be stored. The specification uses two integers from this range to allow special values to be stored. A biased exponent containing all zeros is used to represent subnormal values, which are values with an exponent value of 2^{1-bias} and a mantissa value of $(0 + \beta)$ rather than $(1 + \beta)$. A biased exponent containing all ones indicates either a Not-a-Number (NaN) or an infinity (Inf) if all mantissa bits are zero. The total number of representable exponent values K is

$$K = 2^Z - 2. \tag{4}$$

Adjusting the biased exponent range to allow for the special values, the unsigned integer bits can take integer values in the range $[1, 2046]$. To allow the exponents to be signed, the range is divided in two, which allows the representation of values in the range $[1, 1023]$. This is expressed analytically as

$$\begin{aligned} bias &= 2^Z/2 - 1 \\ &= 2^{Z-1} - 1. \end{aligned} \tag{5}$$

$$\underset{\text{Exponent}}{\begin{Bmatrix} 2^{-1} \\ 2^{0} \\ 2^{1} \end{Bmatrix}} \Rightarrow \underset{\text{Biased}}{\begin{Bmatrix} 1022 \\ 1023 \\ 1024 \end{Bmatrix}} \Rightarrow \underset{\text{Storage}}{\begin{Bmatrix} 01111111110 \\ 01111111111 \\ 10000000000 \end{Bmatrix}}$$

Figure 1: Exponent values, biased exponent values, and storage.

The bias value for an 11 bit exponent is 1023. Examples of exponents and biased exponents are shown in Figure 1. The representable exponents, (α) in our terminology, are the powers of two in the range

$$\alpha \in \{2^{-1022}, 2^{-1021}, \ldots, 2^{1023}\}. \tag{6}$$

The biased exponents e take integer values in the range $[1, 2046]$, or

$$e = \sum_{i=0}^{Z-1} b_i \times 2^i, \tag{7}$$

where b_i is the i-th exponent bit. The desired exponent value *alpha* can then be written as $\alpha = 2^{e-bias}$.

1.1. Mantissa

Consider a corrupted scalar, $\tilde{a}$. For corruption affecting the mantissa, the corruption can be treated algebraically, regardless of the number of bits affected.

$$\begin{aligned} \tilde{a} &= \alpha \times 1.\tilde{\beta} \\ &= \alpha \times (1.\beta \pm 1.\xi) \\ &= \alpha \times 1.\beta \pm \alpha \times 1.\xi \\ &= a \pm \alpha \times 1.\xi \end{aligned} \tag{8}$$

The error term $1.\xi$ belongs to a discrete set of values

$$1.\xi \in \{1.0 + 2^{i-N}\} \text{ for } i = 0, \ldots, N-1. \tag{9}$$

The error introduced from a single bit flip is bounded by analyzing the largest and smallest mantissa perturbations. The smallest possible value for ξ is the smallest power of two that can be stored, e.g., $1 + 2^{-N} = 1 + \epsilon \approx 1.0$. The largest value is then $1 + 2^{-1} = 1.5$. A mantissa error from a single bit flip is bounded by

$$1.0 < 1.\xi \leq 1.5. \tag{10}$$

1.2. Exponent

Suppose a corrupted scalar, $\tilde{a}$, experiences a fault that introduces an error into the exponent bits. An exponent bit flip is really a bit flip in the *biased exponent*. That is

$$\tilde{a} = \tilde{\alpha} \times 1.\beta. \tag{11}$$

Expanding $\tilde{\alpha}$,

$$\begin{aligned} \tilde{\alpha} &= 2^{\tilde{e}-bias} \\ &= 2^{e \pm \eta - bias} \\ &= 2^{e-bias} \times 2^{\pm\eta} \\ &= \alpha \times 2^{\pm\eta}. \end{aligned} \tag{12}$$

Table 3: Statistics Terminology

X_*	A random variable (R.V.).
X_β	R.V. taking values of the mantissa ($1.\beta$).
X_ω	R.V. taking values of the reciprocal of the mantissa ($\frac{1}{1.\beta}$).
X_ξ	R.V. taking values of the absolute error introduced into the mantissa ($1.\xi$).
X_α	R.V. taking values of the true exponent (α).
X_η	R.V. taking values of the error introduced by a bit flip in the biased exponent (2^η).
$\mathbb{E}[X]$	The expected value of X.
$\mathrm{Var}(X)$	The variance of X.
$\mathrm{Cov}(X, Y)$	The covariance.

Hence,

$$\tilde{a} = a \times 2^{\pm\eta} \tag{13}$$

The error term $2^{\pm\eta}$ is key. Recall, the biased exponent e is an unsigned integer value stored in the exponent bits, as shown in Eq. (7). Given a single bit flip in the j-th bit,

$$\begin{aligned}\tilde{e} &= \sum_{i=0}^{Z-1} b_i \times 2^i \pm 2^j \\ &= e \pm \eta.\end{aligned}$$

The multiplicative error term is then $2^{\pm 2^j}$. Given a single bit flip and Z exponent bits, there are $2Z$ possible values for the error term's exponent

$$\pm\eta \in \{\pm 2^0, \pm 2^1, \pm 2^2, \ldots, \pm 2^{Z-1}\}. \tag{14}$$

Recognize that error term's exponent takes values that are positive or negative. A plus indicates a bit that is flipped $0 \to 1$, while a minus indicates a bit was flipped $1 \to 0$. Because the error is multiplicative, the sign of η determines if the scalar's exponent is increased (addition in the exponent) or decreased (subtraction in the exponent). For example, given a scalar with exponent $\alpha = 2^0$, the biased exponent is $e = 1023 = 011\ 1111\ 1111$. A bit flip in the lowest order exponent bit toggles a $1 \to 0$. This introduces an error $2^{-\eta}$ with $\eta = 2^0$, or $\alpha \times 2^{-2^0} = \alpha \times 2^{-1}$.

1.3. Sign

For an error impacting the sign bit,

$$\tilde{a} = -a. \tag{15}$$

2. Model Statistics

Using Eq. (1) and the form presented in Eq. (11), we now analyze statistics for each component and two error measures. The goal is to derive analytic representations of the expected value for discrete operations, and then to determine how the error measures behave when perturbed with a bit flip. Table 3 summarizes the variables and notation used for our statistical analysis.

2.1. Mantissa

The number of possible values representable using N bits is $M = 2^N$. Recognize that the set of all mantissa values is

$$\begin{aligned} 1.\beta &\in \{1.0, 1+\epsilon, 1+2\epsilon, \ldots, 1+(M-1)\epsilon\} \\ 1.\beta &\in \left\{ 1.0 + \sum_{j=i}^{M} 2^{-N} \right\} \text{ for } i = 1, \ldots, M \\ 1.\beta &\in \{1.0 + 2^{-N}(M-i)\} \text{ for } i = 1, \ldots, M \end{aligned} \tag{16}$$

Let X_β be a discrete random variable that takes values from Eq. (16) with equally likely probability. The expected value of the mantissa is

$$\begin{aligned} \mathbb{E}[X_\beta] &= \frac{1}{M} \sum_{i=1}^{M} \left[1 + \sum_{j=i}^{M} 2^{-N} \right] \\ &= \frac{1}{M} \sum_{i=1}^{M} \left[1 + 2^{-N}(M-i) \right] \\ &= \frac{1}{M} \sum_{i=1}^{M} \left[1 + 2^{-N}2^{N} - 2^{-N}i \right] \\ &= \frac{1}{M} \sum_{i=1}^{M} \left[2 - 2^{-N}i \right] \\ &= \frac{1}{M} \left[2M - 2^{-N} \sum_{i=1}^{M} i \right] \\ &= \frac{1}{M} \left[2M - 2^{-N} \frac{M(M+1)}{2} \right] \\ &= 2 - 2^{-N} \frac{M+1}{2} \\ &= 2 - \frac{1 + 2^{-N}}{2} \\ &= 2 - \frac{1}{2} - \frac{2^{-N}}{2} \\ &= 1.5 - \frac{\epsilon}{2} \end{aligned} \tag{17}$$

Alternatively, consider the continuous interval $[1, 2]$, the expected value is $\int_1^2 x \mathrm{d}x = \frac{1}{2}x^2|_1^2 = 1.5$.
The variance of the mantissa is defined as

$$\mathrm{Var}(X_\beta) = \mathbb{E}\left[X_\beta^2\right] - \mathbb{E}[X_\beta]^2. \tag{18}$$

The expected value of X_β^2 is

$$\begin{aligned}
\mathbb{E}\big[X_\beta^2\big] &= \frac{1}{M}\sum_{i=1}^{M}\left[1+\sum_{j=i}^{M}2^{-N}\right]^2 \\
&= \frac{1}{M}\sum_{i=1}^{M}\left[2-2^{-N}i\right]^2 \\
&= \frac{1}{M}\sum_{i=1}^{M}\left[4-2^{2-N}i+2^{-2N}i^2\right] \\
&= \frac{1}{M}\left[4M-2^{2-N}\frac{M(M+1)}{2}\right. \\
&\quad \left.+\,2^{-2N}\frac{M(M+1)(2M+1)}{6}\right] \\
&= 4-2^{1-N}(M+1)+2^{-2N}\frac{(M+1)(2M+1)}{6} \\
&= 4-2-2^{1-N}+\frac{2^{-2N}(2M^2+3M+1)}{6} \\
&= 2-2^{1-N}+\frac{2+3\times 2^{-N}+2^{-2N}}{6} \\
&= 2-2^{1-N}+\frac{2}{6}+\frac{3}{6}2^{-N}+\frac{1}{6}2^{-2N} \\
&= 2-2^{1-N}+\frac{1}{3}+\frac{1}{2}2^{-N}+\frac{1}{6}2^{-2N} \\
&= \frac{7}{3}-2\epsilon+\frac{1}{2}\epsilon+\frac{1}{6}\epsilon^2 \\
&= \frac{7}{3}-\frac{3}{2}\epsilon+\frac{1}{6}\epsilon^2, \qquad (19)
\end{aligned}$$

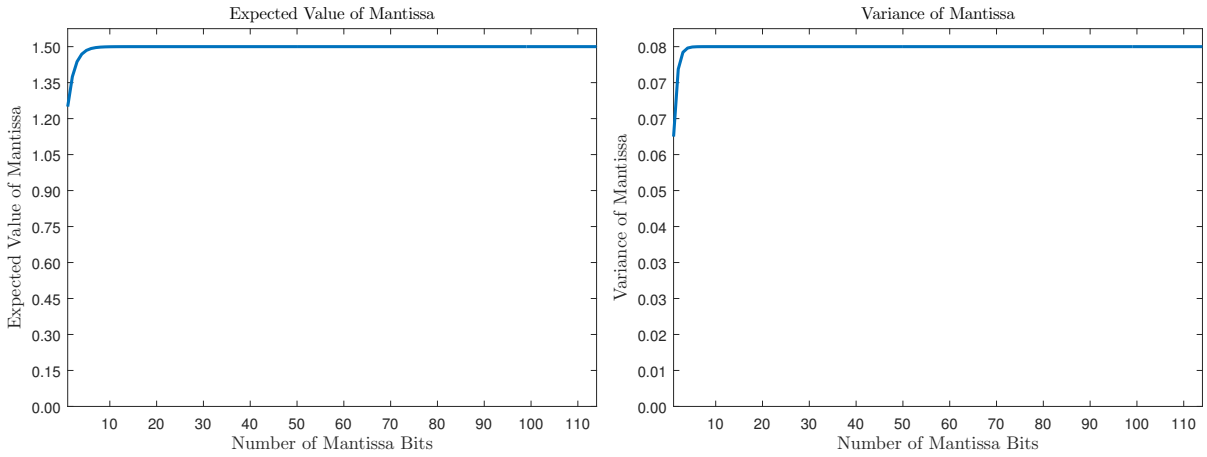
and the square of the expectation of $1.\beta$ is

$$\begin{aligned}
\mathbb{E}[X_\beta]^2 &= \left(1.5-\frac{\epsilon}{2}\right)^2 \\
&= \frac{9}{4}-\frac{3}{2}\epsilon+\frac{1}{4}\epsilon^2. \qquad (20)
\end{aligned}$$

Substituting Eqs. (19) and (20) into Eq.(18) yields the variance of the mantissa

$$\begin{aligned}
\mathrm{Var}(X_\beta) &= \mathbb{E}\big[X_\beta^2\big]-\mathbb{E}[X_\beta]^2 \\
&= \frac{1}{12}-\frac{1}{12}\epsilon^2. \qquad (21)
\end{aligned}$$

Recognize that Eq. (21) is an increasing function that rapidly converges to $\frac{1}{12}$. Figure 2a plots the expected value of the mantissa as a function of the number of bits used to store the mantissa, while Figure 2b shows the variance as function of the number of bits used.



(a) Expected value of the mantissa given N bits.

(b) Variance of the mantissa given N bits.

Figure 2: Expected value and variance of the an N-bit mantissa.

2.2. Mantissa Errors

Equation (9) shows the set of unscaled mantissa single bit errors. Let X_ξ be a discrete random variable taking values from Eq. (9) with equally likely probability. The expected value for an N-bit mantissa is

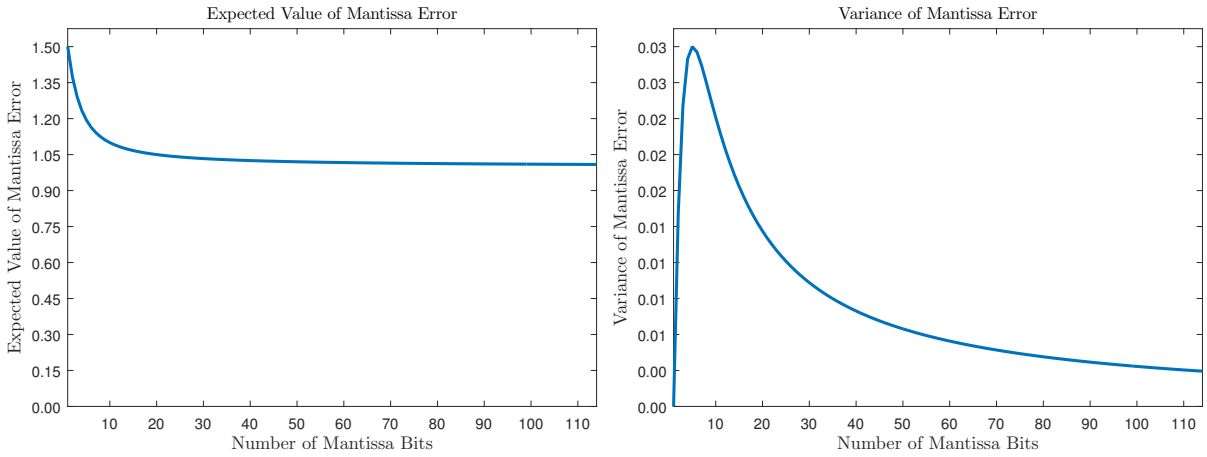
$$\begin{aligned}
\mathbb{E}[X_\xi] &= \frac{1}{N}\sum_{i=1}^{N}\left(1+2^{-i}\right) \\
&= \frac{1}{N}\left(N+\sum_{i=1}^{N}2^{-i}\right) \\
&= \frac{1}{N}\left(N+\sum_{i=0}^{N-1}ar^{i}\right);\ \text{for } a=r=1/2 \\
&= \frac{1}{N}\left(N+1.0-2^{-N}\right) \\
&= 1+N^{-1}-\frac{\epsilon}{N}. \qquad (22)
\end{aligned}$$

The variance of a mantissa error is

$$\mathrm{Var}(X_\xi) = \mathbb{E}\left[X_\xi^2\right] - \mathbb{E}[X_\xi]^2. \qquad (23)$$

The expected value of X_ξ^2 is

$$\begin{aligned}
\mathbb{E}\left[X_\xi^2\right] &= \frac{1}{N}\sum_{i=1}^{N}\left(1+2^{-i}\right)^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(1+2^{1-i}+2^{-2i}\right) \\
&= \frac{1}{N}\left(N+2[1-\epsilon]+\frac{1}{3}(1-\epsilon^2)\right) \\
&= \frac{1}{N}\left(N+2-2\epsilon+\frac{1}{3}-\frac{\epsilon^2}{3}\right) \\
&= 1+\frac{7}{3N}-\frac{2\epsilon}{N}-\frac{\epsilon^2}{3N}, \qquad (24)
\end{aligned}$$



(a) Expected value of the absolute error.

(b) Variance of the absolute error.

Figure 3: Expected value and variance of the absolute error from a single bit flip in an N-bit mantissa.

and the squared expected value is

$$\begin{aligned}\mathbb{E}[X_\xi]^2 &= \left[1 + \frac{1}{N} - \frac{\epsilon}{N}\right]^2 \\ &= 1 + 2\frac{1-\epsilon}{N} + \frac{(1-\epsilon)^2}{N^2} \\ &= 1 + \frac{2}{N} + \frac{1}{N^2} - \frac{2\epsilon}{N} - \frac{2\epsilon}{N^2} + \frac{\epsilon^2}{N^2}.\end{aligned} \tag{25}$$

The variance is obtained by substituting Eqs. (24) and (25) into Eq. (23)

$$\mathrm{Var}(X_\xi) = \frac{1}{3N} + \frac{2\epsilon}{N^2} - \frac{1}{N^2} - \frac{\epsilon^2}{3N} - \frac{\epsilon^2}{N^2}. \tag{26}$$

Figure 3 plots the expected error and the variance of the error that is introduced if a bit is flipped in the mantissa. Figure 3a plots the expected value of X_ξ as a function of the number of bits used to represent the mantissa. Figure 3b plots the variance as a function of the number of mantissa bits (N). Note the differences in the y-axes.

Recognize that when $N = 1$, $\mathrm{Var}(X_\xi) = 0$, and as $N \to \infty$, the variance increases before tending towards 0. This increase may be seen by inspecting the derivative with respect to N. The variance reaches a maximum at $N = 5$. As $N \gg 5$, the variance decreases towards zero, and the expected value converges to 1.0. The important fact is that the expected mantissa error is approximately 1.0, and the variance is roughly zero. Because the variance is small, and the mean converges to one, we consider the mantissa errors to be "well behaved".

2.3. Exponent

Let X_α be a discrete random variable taking values from Eq. (6). The number of representable exponents is K (see Eq. (4)). Recall, because the exponent bits store an unsigned integer, K represents an unsigned

count of representable exponents. The expected value of X_α is

$$\begin{aligned}
\mathbb{E}[X_\alpha] &= \frac{1}{K}\sum_{i=1}^{K} 2^{i-bias} \\
&= \frac{1}{K}2^{-bias}\sum_{i=1}^{K} 2^{i} \\
&= \frac{1}{K}2^{-bias}\left[\sum_{i=0}^{K} 2^{i} - 1\right] \\
&= \frac{1}{K}2^{-bias}\left[2^{K+1} - 2\right] \\
&= \frac{1}{K}\left[2^{K+1-bias} - 2^{1-bias}\right]
\end{aligned}$$

Recognize from Eq. (5) that K can be written as a function of the $bias$, $K = 2 \times bias$. The expected value may then be expressed as

$$\begin{aligned}
\mathbb{E}[X_\alpha] &= \frac{1}{K}\left[2^{bias+1} - 2^{1-bias}\right] \\
&= \frac{1}{bias}\left[2^{bias} - 2^{-bias}\right]. \qquad (27)
\end{aligned}$$

Recognize the expected value is dominated by 2^{bias}, yielding, an approximation

$$\mathbb{E}[X_\alpha] \approx \frac{2^{bias}}{bias}.$$

Comparing Eq. (27) to the continuous expected value on the interval $[2^{-1022}, 2^{1023}]$,

$$\begin{aligned}
\int_{-1022}^{1023} 2^x \mathrm{d}x &= \left.\frac{2^x}{\ln 2}\right|_{-1022}^{1023} \\
&= \frac{2^{bias}}{\ln 2}. \qquad (28)
\end{aligned}$$

The continuous expected value provides an upper bound on the discrete approximation, that is, $\frac{2^{bias}}{bias} < \frac{2^{bias}}{\ln 2}$.

The variance of the exponent is

$$\mathrm{Var}(X_\alpha) = \mathbb{E}\left[X_\alpha^2\right] - \mathbb{E}[X_\alpha]^2 \qquad (29)$$

with expected values

$$\begin{aligned}
\mathbb{E}\left[X_\alpha^2\right] &= \frac{1}{K}\sum_{i=1}^{K}\left(2^{i-bias}\right)^2 \\
&= \frac{2^{-2bias}}{K}\sum_{i=1}^{K}2^{2i} \\
&= \frac{2^{-K}}{K}\sum_{i=1}^{K}4^{i} \\
&= \frac{2^{-K}}{K}\left[\frac{4^{K+1}-1}{3}-1\right] \\
&= \frac{2^{-K}}{3K}\left[4^{K+1}-4\right] \\
&= \frac{2^{2-K}}{3K}\left[2^{2K}-1\right] \\
&= \frac{2^{K+2}}{3K}-\frac{2^{2-K}}{3K}.
\end{aligned} \tag{30}$$

$$\begin{aligned}
\mathbb{E}[X_\alpha]^2 &= \left[\frac{1}{K}\left(2^{K+1-bias}-2^{1-bias}\right)\right]^2 \\
&= \frac{1}{K^2}\left[2^{K+2}-2^3+2^{2-K}\right].
\end{aligned} \tag{31}$$

Substituting Eqs. (30) and (31) in Eq. (29) yields the variance

$$\begin{aligned}
\mathrm{Var}(X_\alpha) &= \mathbb{E}\left[X_\alpha^2\right]-\mathbb{E}[X_\alpha]^2 \\
&= \frac{2^{K+2}}{3K}-\frac{2^{2-K}}{3K}-\frac{1}{K^2}\left[2^{K+2}-2^3+2^{2-K}\right] \\
&= \frac{2^{K+2}}{3K}-\frac{2^{K+2}}{K^2}-\frac{2^{2-K}}{3K}+\frac{2^3}{K^2}-\frac{2^{2-K}}{K^2}.
\end{aligned} \tag{32}$$

Recognize the final three terms in Eq. (32) are small. The variance of X_α will be dominated by

$$O\left(\mathrm{Var}(X_\alpha)\right) = \frac{2^{K+2}}{3K}-\frac{2^{K+2}}{K^2}. \tag{33}$$

To understand the growth of Eq. (33) we analyze the limit as K goes to infinity

$$O\left(\mathrm{Var}(X_\alpha)\right) = \lim_{k\to\infty^+}\frac{2^{K+2}K-2^{K+2}3}{3K^2}.$$

With two applications of L'Hôpital's rule, we obtain

$$O\left(\mathrm{Var}(X_\alpha)\right) = \frac{2^{K+2}\ln^2(2)(K-3)+2^{K+2}\ln(2)\times 2}{6}. \tag{34}$$

The limit as $K\to\infty$ is ∞, and the dominant term in the summation is $2^{K+2}\ln(2)/3$, or $2^{2\times bias+2}\ln(2)/3$. Figure 4 plots the upper bound on the variance as a function of the number of bits used to represent the exponent, as well as the variance. The variance is large, which means that values are not clustered near the mean. The standard deviation (σ) is the square root of the variance. Recognize that $2^{2\times bias+2}=(2^{bias+1})^2$, hence $\sigma = 2^{bias+1}$. That is, σ is approximately the mean. Compare this to the exponential distribution, where $\mathbb{E}\left[e^{-\lambda x}\right]=1/\lambda$, and the variance is $(1/\lambda)^2$. That is, our model follows the similar continuous exponential distribution, with the variance approximately the square of the mean.

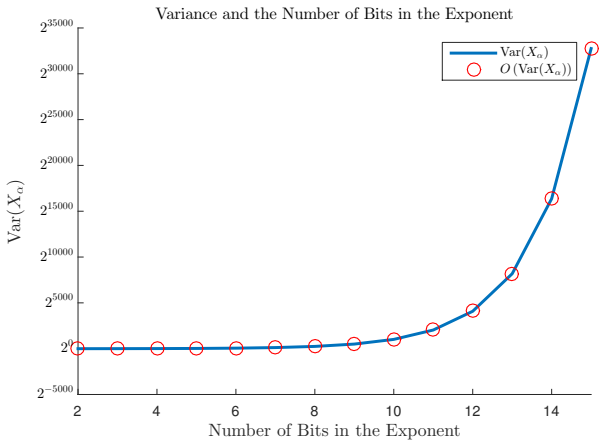


Figure 4: Variance of the exponent and the dominant term in the growth of the variance as a function of the number of exponent bits.

2.4. Exponent by Range

The expected value of the exponent falls into two broad ranges: values with positive exponents, i.e., the biased exponent is larger than the bias, and negative exponents, where the biased exponent is smaller than the bias. Partitioning α in two sets $\alpha^+ \in \{2^0, 2^1, \ldots 2^{bias}\}$ and $\alpha^- \in \{2^{-1}, \ldots, 2^{1-bias}\}$. We assign 2^0 to the positive set, because the resulting value is not a fraction.

Let X_{α^+} be a discrete random variable that takes values from the set of positive exponents with equally likely probability, and let X_{α^-} be a discrete random variable that takes values from the negative set of exponents with equally likely probability. The expected value of the positive exponents is

$$\begin{aligned}\mathbb{E}[X_{\alpha^+}] &= \frac{1}{bias+1}\sum_{i=0}^{bias+1} 2^i \\ &= \frac{2^{bias+1}-1}{bias+1}.\end{aligned} \tag{35}$$

The expected value of the negative exponents is

$$\begin{aligned}\mathbb{E}[X_{\alpha^-}] &= \frac{1}{bias-1}\sum_{i=1}^{bias-1} 2^{-i} \\ &= \frac{1-2^{1-bias}}{bias-1}.\end{aligned} \tag{36}$$

Recall that Z is the number of bits used to store biased exponents. To implement the specification, Z must be at least 2, because two values are reserved (zero and $2^Z - 1$). This requires at least 2 bits, otherwise, no exponents are representable. To store a negative exponent,

$$Z \geq 3, \tag{37}$$

because with $Z = 2$, only exponents with zero and $+1$ are representable. Hence, with $Z \geq 3$, $bias \geq 3$ and the term $\frac{2^{1-bias}}{bias-1} < 1.0$. The expected value can then be bounded as

$$\mathbb{E}[X_{\alpha^-}] < \frac{1}{bias-1}. \tag{38}$$

The point of deriving Equations (35) and (36) will become clearer when we apply these findings to draw high-level conclusions about injecting a bit flip into the representation of a floating point value. Specifically,

$\mathbb{E}[X_\alpha]$ will continue to be a dominant term in most errors models. Once we have analyzed specific error models, we will show how a small subset of the total possible bit flips can contribute excessively larger error relative to all other bit flips.

2.5. Exponent Errors

The expected error introduced from a bit flip in the exponent is the expected value of 2^η. Let X_η be a discrete random variable taking values from Eq. (14) with equally likely probability. Partition this set into positive and negative subsets, and let X_{η^+} be a R.V. that takes values from the positive set $\{2^0, \ldots, 2^{Z-1}\}$ and let X_{η^-} be a R.V. that takes values from the negative set $\{-2^0, \ldots, -2^{Z-1}\}$.

$$\begin{aligned}\mathbb{E}[X_\eta] &= \frac{1}{2Z}\sum_{i=0}^{Z-1} 2^{2^i} + \frac{1}{2Z}\sum_{i=0}^{Z-1} 2^{-2^i} \\ &= \frac{1}{2}\left[\mathbb{E}\left[X_{\eta^+}\right] + \mathbb{E}\left[X_{\eta^-}\right]\right]. \end{aligned} \tag{39}$$

The form 2^{2^n} has been studied extensively, as $F_n = 2^{2^n} + 1$ is a Fermat number, which are studied in relation to prime numbers. No closed form exists for the sum of the Fermat numbers or of its reciprocals.

2.5.1. Positive Exponent Set

The expected value of the positive set $\mathbb{E}\left[X_{\eta^+}\right]$ can be bounded recognizing that

$$\sum_{i=0}^{Z-1} 2^{2^i} < \sum_{j=0}^{2^{Z-1}} 2^j = 2^{2^{Z-1}+1} - 1. \tag{40}$$

Eq. (40) exploits the relation $2^{G+1} - 1 = \sum_{k=0}^{G} 2^k$, where G is a positive integer. A lower bound may be constructed by recognizing that

$$2^{2^{Z-1}} < \sum_{i=0}^{Z-1} 2^{2^i}, \text{ for } Z \geq 2. \tag{41}$$

The inequality in Eq. (41) is strict, because $Z \geq 2$. This gives a bound of

$$2^{2^{Z-1}} < \sum_{i=0}^{Z-1} 2^{2^i} < 2^{2^{Z-1}+1} - 1.$$

The expected value is then bounded by

$$\begin{aligned}\frac{2^{2^{Z-1}}}{Z} < \mathbb{E}\left[X_{\eta^+}\right] < \frac{2^{2^{Z-1}+1} - 1}{Z}, \\ \frac{2^{bias+1}}{Z} < \mathbb{E}\left[X_{\eta^+}\right] < \frac{2^{bias+2} - 1}{Z}. \end{aligned} \tag{42}$$

Figure 5 plots the expected value and its lower and upper bounds. The upper bound is constructed by summing all positive powers of two, and using this sum as a strict upper bound. The upper bound is an approximation of the continuous expected value $\int_0^{bias+1} 2^x \mathrm{d}x$, which is a good approximation of $\mathbb{E}[X_\alpha]$, as shown in Eq. (28). This shows then when summing exponentially distributed values, the large values will dominate the expected value, $\mathbb{E}\left[X_{\eta^+}\right] \approx 2^{bias}$, which is approximately the mean of the exponents $\mathbb{E}[X_\alpha]$ (see Eq. (27)).

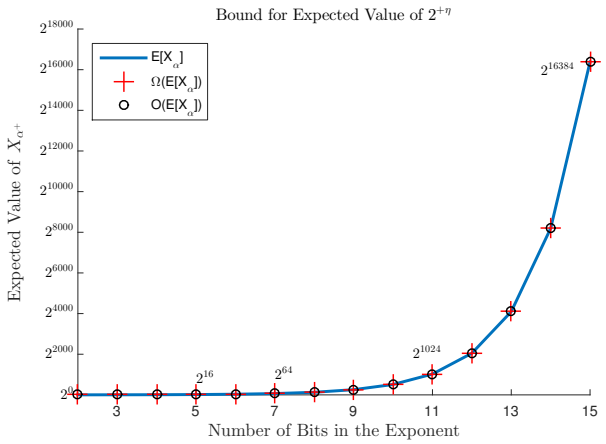


Figure 5: Lower and upper bounds for the expected value of a positive exponent error $\left(\mathbb{E}\left[X_{\eta^+}\right]\right)$.

2.5.2. Negative Exponent Set

The second set of potential exponent errors fall in the class of negative values for η. The expected value of interest is

$$\mathbb{E}\left[X_{\eta^-}\right] = \frac{1}{Z}\sum_{i=0}^{Z-1} 2^{-2^i}.$$

The sum has been shown to be less than one [1]. This gives an upper bound of

$$\mathbb{E}\left[X_{\eta^-}\right] < \frac{1}{Z}. \tag{43}$$

2.5.3. Exponent Error Expected Value

The expected value of the exponent errors is obtained by substituting back into Eq. (39). An upper bound on the expected value is obtained by substituting Eq. (40) for $\mathbb{E}\left[X_{\eta^+}\right]$

$$\begin{aligned}\mathbb{E}[X_\eta] &< \frac{1}{2}\left[\frac{2^{2^{Z-1}+1}-1}{Z} + \frac{1}{Z}\right] \\ &< \frac{1}{2}\left[\frac{2^{bias+2}}{Z}\right] \\ &< \frac{2^{bias+1}}{Z}.\end{aligned}$$

A lower bound on the expected value is obtained by substituting Eq. (42) for $\mathbb{E}\left[X_{\eta^+}\right]$

$$\begin{aligned}\mathbb{E}[X_\eta] &> \frac{1}{2}\left[\frac{2^{2^{Z-1}}}{Z} + \frac{1}{Z}\right] \\ &> \frac{1}{2}\left[\frac{2^{bias+1}}{Z}\right] \\ &> \frac{2^{bias}}{Z}.\end{aligned}$$

Bounding the exponent errors as

$$\frac{2^{bias}}{Z} < \mathbb{E}[X_\eta] < \frac{2^{bias+1}}{Z} \tag{44}$$

Clearly, the expected value of an exponent error is dominated by the positive exponent set.

2.6. Summary of Scalar Statistics

We summarize the statistics for each component of the model in Table 4. These are models for the correct and faulty components of the IEEE-754 representation when perturbed by a single bit flip. In § 4, we use these terms to compose statistics for each error measure, and then extend this analysis to specific operations. Key observations are: 1) Mantissa errors are well-behaved, having an expected error of approximately one, with small variance. 2) The expected exponent error, considering both positive and negative exponents, is approximately the expected value of the exponent range. If a bit flips in the exponent $0 \to 1$ (2^{η^+}), then the expected error is approximately the expected value of the *entire* exponent range. If a bit flips in the exponent $1 \to 0$ the expected error is less than one. Practically, this means that if bits are flipped at random and each bit is equally likely to be a one or zero, then the expected error introduced will be the expected value of the entire (or positive) exponents.

Table 4: Expected value for components of faulty scalars.

Term	R.V.	$\mathbb{E}[\cdot]$	Ref.
$1.\beta$	X_β	$1.5 - \frac{\epsilon}{2}$	Eq. (17)
$1.\xi$	X_ξ	$1 + N^{-1} - \frac{\epsilon}{N}$	Eq. (22)
α	X_α	$\frac{1}{bias}\left[2^{bias} - 2^{-bias}\right]$	Eq. (27)
Pos. α	X_{α^+}	$\frac{2^{bias+1}-1}{bias+1}$	Eq. (35)
Neg. α	X_{α^-}	$< \frac{1}{bias-1}$	Eq. (38)
2^η	X_η	$\frac{2^{bias}}{Z} < \mathbb{E}[X_\eta] < \frac{2^{bias+1}}{Z}$	Eq. (44)
2^{η^+}	X_{η^+}	$\frac{2^{bias+1}}{Z} < \mathbb{E}\left[X_{\eta^+}\right] < \frac{2^{bias+2}-1}{Z}$	Eq. (42)
2^{η^-}	X_{η^-}	$< \frac{1}{Z}$	Eq. (43)

3. Model Error Measures

We now model the absolute and relative errors for a scalar a represented following Eq. (2). The absolute error ($error_{\text{abs}}$) presents the actual error that a bit flip introduces

$$error_{\text{abs}} = |a - \tilde{a}|.$$

The relative error ($error_{\text{rel}}$) indicates how large an error is, relative to the correct value

$$error_{\text{rel}} = \frac{|a - \tilde{a}|}{|a|}.$$

3.1. Mantissa

Equation (8) presents the form of $\tilde{a}$. The absolute error is

$$\begin{aligned} |a - \tilde{a}| &= |a - a \pm \alpha \times 1.\xi| \\ &= |\alpha \times 1.\xi|, \end{aligned} \tag{45}$$

Table 5: Error measures for scalars.

Location	Absolute Error	Ref	Relative Error	Ref
Mantissa	$\lvert\alpha \times 1.\xi\rvert$	Eq. (45)	$\frac{1.\xi}{1.\beta}$	Eq. (46)
Exponent	$\lvert a(1-2^{\eta})\rvert$	Eq. (47)	$\lvert 1-2^{\eta}\rvert$	Eq. (48)
Sign	$\lvert 2a\rvert$	Eq. (49)	2	Eq. (50)

and the relative error is

$$\begin{aligned}\frac{|a-\tilde{a}|}{|a|} &= \frac{|\alpha \times 1.\xi|}{|\alpha \times 1.\beta|} \\ &= \frac{1.\xi}{1.\beta}.\end{aligned} \tag{46}$$

Note, we may drop the absolute value as the sign is the same between both the perturbed scalar and the correct.

3.2. Exponent

Equation (13) presents the form of $\tilde{a}$. The absolute error is

$$\begin{aligned}|a-\tilde{a} &= |a - a \times 2^{\eta}| \\ &= |a(1-2^{\eta})|,\end{aligned} \tag{47}$$

and the relative error is

$$\begin{aligned}\frac{|a-\tilde{a}|}{|a|} &= \frac{|a(1-2^{\eta})|}{|a|} \\ &= |1-2^{\eta}|.\end{aligned} \tag{48}$$

3.3. Sign

Equation (15) presents the form of $\tilde{a}$. The absolute error is

$$\begin{aligned}|a-\tilde{a} &= |a-(-a)| \\ &= |2a|,\end{aligned} \tag{49}$$

and the relative error is

$$\begin{aligned}\frac{|a-\tilde{a}|}{|a|} &= \frac{|2a|}{|a|} \\ &= 2.\end{aligned} \tag{50}$$

We summarize the scalar error measures in Table 5.

4. Scalar Expected Error Measures

Having shown statistics for components of the IEEE-754 model, we explore the expected value of the absolute and relative error given a bit flip in an IEEE-754 floating point scalar.

4.1. Expected Relative Error for a Scalar

4.1.1. Mantissa

We now compute the expected relative error for a scalar with a mantissa error, $\mathbb{E}\left[\frac{1.\xi}{1.\beta}\right]$. The R.V. X_ξ was defined and analyzed in Eq. (22). Let X_ω be a discrete random variable taking values from

$$\frac{1}{1.\beta} \in \left\{ \frac{1}{1.0 + 2^{-N}(M-i)} \right\} \text{ for } i = 1, \ldots, M,$$

with equally likely probability. The expected value is

$$\begin{aligned} \mathbb{E}\left[\frac{1.\xi}{1.\beta}\right] &= \mathbb{E}[X_\xi X_\omega] \\ &= \text{Cov}(X_\xi, X_\omega) + \mathbb{E}[X_\xi]\mathbb{E}[X_\omega], \end{aligned} \tag{51}$$

where $\text{Cov}(u, v)$ is the covariance. The covariance is defined to be

$$\begin{aligned} \text{Cov}(u, v) &= \mathbb{E}[(u - \mathbb{E}[u])(v - \mathbb{E}[v])] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \end{aligned}$$

We begin with

$$\begin{aligned} \text{Cov}(X_\xi, X_\omega) &= \mathbb{E}[(X_\xi - \mathbb{E}[X_\xi])(X_\omega - \mathbb{E}[X_\omega])] \\ &= \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\left(1 + 2^{-i} - \mathbb{E}[X_\xi]\right) (X_\omega - \mathbb{E}[X_\omega]) \right] \\ &= \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\left(1 + 2^{-i} - \left(1 + \frac{1}{N} - \frac{\epsilon}{N}\right)\right) (X_\omega - \mathbb{E}[X_\omega]) \right] \\ &= \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\left(2^{-i} - \frac{1}{N} + \frac{\epsilon}{N}\right) (X_\omega - \mathbb{E}[X_\omega]) \right] \\ &= \frac{1}{NM} \sum_{i=1}^{N} \left(2^{-i} + \frac{\epsilon - 1}{N}\right) \sum_{j=1}^{M} (X_\omega - \mathbb{E}[X_\omega]) \\ &= \frac{1}{NM} \left[1 - 2^{-N} + \frac{\epsilon - 1}{N} \times N\right] \sum_{j=1}^{M} (X_\omega - \mathbb{E}[X_\omega]) \\ &= \frac{1}{NM} \left[1 - \epsilon + \epsilon - 1\right] \sum_{j=1}^{M} (X_\omega - \mathbb{E}[X_\omega]) \\ &= 0. \end{aligned} \tag{52}$$

Therefore, $\mathbb{E}\left[\frac{1.\xi}{1.\beta}\right] = \mathbb{E}[X_\xi]\mathbb{E}[X_\omega]$. The expected value of X_ξ is shown in Eq. (22). The expected value of the reciprocal of the mantissa is

$$\mathbb{E}[X_\omega] = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{1 + 2^{-N}(M-i)}.$$

Recognize this as a left Riemann sum approximation of the integral

$$\begin{aligned} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{1 + 2^{-N}(M-i)} &= \int_0^1 \frac{1}{1+x} \mathrm{d}x \\ &= \ln(2). \end{aligned} \tag{53}$$

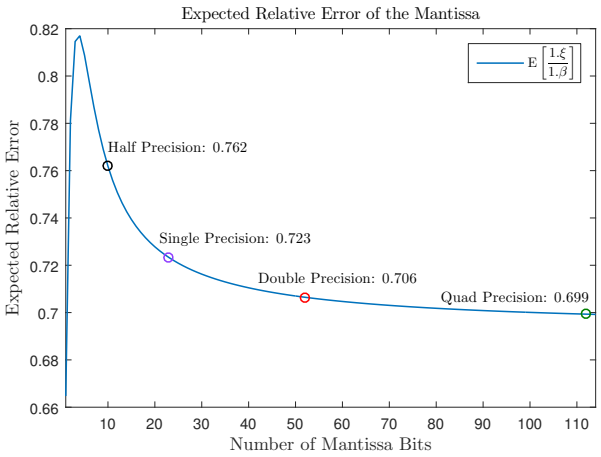


Figure 6: Expected relative error of the mantissa given a single bit flip.

The error of a left Riemann sum is

$$\begin{aligned} E_L &= \frac{b-a}{2} \times \max(f'(c)) \times \Delta x \\ &= -\frac{1}{2}\frac{(1-0)^2}{M}, \end{aligned}$$

where $\Delta x = \frac{b-a}{M}$. Recognize that for $f'(x) = -\frac{1}{(1+x)^2}$, the maximum value over the interval $[0, 1]$ is 1 resulting in the approximation error

$$\begin{aligned} E_L &= -\frac{1}{2}\frac{1}{M} \\ &= -\frac{\epsilon}{2}. \end{aligned}$$

Note $M = 2^N$, hence $1/M = \epsilon$. The expected value of the reciprocal of the mantissa is Eq. (53) plus the approximation error

$$\mathbb{E}[X_\omega] = \ln(2) - \frac{\epsilon}{2}. \tag{54}$$

Substituting Eqs. (54), (52), and (22) into Eq. (51) yields the expected value of the relative error

$$\begin{aligned} \mathbb{E}\left[\frac{1.\xi}{1.\beta}\right] &= \mathbb{E}[X_\xi]\mathbb{E}[X_\omega] \\ &= \left[1 + \frac{1}{N} - \frac{\epsilon}{N}\right] \times \left[\ln(2) - \frac{\epsilon}{2}\right] \\ &= \ln(2)\left[1 + \frac{1}{N}\right] - \frac{2\ln(2) + N + 1}{2N}\epsilon + \frac{1}{2N}\epsilon^2. \end{aligned} \tag{55}$$

Given that $N > 0$, then Eq. (55) obtains a maximum at $N = 4$. Figure 6 plots the expected relative error as a function of the number of bits used in the mantissa.

A key point of Eq. (55) is that the expected relative error of a scalar that experiences a bit flip in the mantissa is less than one. In prior work, we loosely bound the mantissa errors by incrementing the exponent. That is, no bit flip in the mantissa could ever generate an error larger than twice the scalar's exponent. Clearly, our prior upper bound is an extreme case, while the expected relative error is much smaller.

4.1.2. Statistical Independence

Theorem 4.1. *The error introduced by a bit flip in the mantissa ($x_\xi = 1.\xi$) is independent of the mantissa value ($x_\beta = 1.\beta$).*

Proof 4.1. *The joint probability mass function $p_{X_\xi, X_\beta}(x_\xi, x_\beta) = P(X_\xi = x_\xi \text{ and } X_\beta = x_\beta)$. The marginal probability mass function for the error introduced from a single bit flip in the mantissa is*

$$p_{X_\xi}(x_\xi) = \sum_{x_\beta} p(x_\xi, x_\beta),$$

and the marginal probability mass function for the mantissa is

$$p_{X_\beta}(x_\beta) = \sum_{x_\xi} p(x_\xi, x_\beta).$$

The probability that $X_\xi = x_\xi$ is the probability of experiencing a bit flip in the i-th bit, which is $\frac{1}{N}$. The probability of having a specific mantissa is $X_\beta = x_\beta$, which is $\frac{1}{M}$.

The joint probability of experiencing a specific mantissa error (x_ξ) given a specific mantissa (x_β) is $p(x_\xi, x_\beta) = \frac{1}{NM}$. Two random variables are independent if $p_{X_\xi, X_\beta}(x_\xi, x_\beta) = p_{X_\xi}(x_\xi) p_{X_\beta}(x_\beta)$.

$$p_{X_\xi}(x_\xi) = \sum_{x_\beta} p(x_\xi, x_\beta) = \sum_{i=1}^{M} \frac{1}{NM} = \frac{M}{NM} = \frac{1}{N}$$

$$p_{X_\beta}(x_\beta) = \sum_{x_\xi} p(x_\xi, x_\beta) = \sum_{i=1}^{N} \frac{1}{NM} = \frac{N}{NM} = \frac{1}{M}$$

$$p_{X_\xi, X_\beta}(x_\xi, x_\beta) = \frac{1}{NM} = p_{X_\xi}(x_\xi) p_{X_\beta}(x_\beta)$$

□

The proof for the reciprocal of the mantissa is similar and is omitted. Theorem 4.1 is powerful, because it means that the relative error introduced from a bit flip in the mantissa does not depend on the mantissa value. This arises because the error $1.\xi$ is $\pm$, e.g., see Eq. (8). Regardless of whether the bit flipped is and 1 to 0 or 0 to 1, the absolute error is the same, i.e., $|\pm 1.\xi| = 1.\xi$. This implies that Figure 6 predicts the expected relative error for a mantissa bit flip for all values represented using the IEEE-754 specification.

4.1.3. Exponent

We analyze the expectation of the relative error of the exponent $|1 - 2^\eta|$ in two ways. First, recognize that the expected value of the relative error is $\mathbb{E}[2^\eta - 1] = \mathbb{E}[X_\eta] - 1$. This is not very informative, given that the expected value of η is dominated by the positive exponents. That is,

$$\frac{2^{bias}}{Z} < \mathbb{E}[2^\eta - 1] < \frac{2^{bias+1}}{Z} \tag{56}$$

Instead, consider the error term broken into positive and negative exponent sets, as shown in Eq. (39). The expected relative error given a negative exponent is

$$\mathbb{E}\big[|1 - X_{\eta^-}|\big] = |1 - \mathbb{E}\big[X_{\eta^-}\big]|,$$

which is bounded by

$$|1 - \mathbb{E}\big[X_{\eta^-}\big]| < 1. \tag{57}$$

The expected relative error for an error creating a positive exponent is $\mathbb{E}[2^{+\eta} - 1]$, which lies within the same bound as $\mathbb{E}\big[X_{\eta^+}\big]$.

$$\frac{2^{bias+1}}{Z} < \mathbb{E}\Big[2^{\eta^+} - 1\Big] < \frac{2^{bias+2} - 1}{Z}. \tag{58}$$

4.2. Expected Absolute Error for a Scalar

4.2.1. Mantissa

Let X_ξ be a discrete random variable taking values from Eq. (9). We break α into two ranges, as we did in § 2.4. X_α is a discrete random variable taking values from Eq. (6), and X_{α^+} and X_{α^-} are discrete random variables taking values from the positive and negative exponent sets. The expected value of the absolute error given a mantissa bit flip is

$$\begin{aligned}\mathbb{E}[\alpha \times 1.\xi] &= \mathbb{E}[X_\alpha X_\xi] \\ &= \text{Cov}(X_\xi, X_\alpha) + \mathbb{E}[X_\xi]\mathbb{E}[X_\alpha]. \end{aligned} \tag{59}$$

The expected value of X_ξ is shown in Eq. (22), and the expected value of X_α is shown in Eq. (27). The covariance will be zero,

$$\begin{aligned}\text{Cov}(X_\xi, X_\alpha) &= \mathbb{E}[(X_\xi - \mathbb{E}[X_\xi])(X_\alpha - \mathbb{E}[X_\alpha])] \\ &= 0, \end{aligned} \tag{60}$$

because $\mathbb{E}[(X_\xi - \mathbb{E}[X_\xi])] = 0$, as shown in Eq. (52). Substituting Eq. (60) into Eq. (59),

$$\begin{aligned}\mathbb{E}[\alpha \times 1.\xi] &= \mathbb{E}[X_\xi]\mathbb{E}[X_\alpha] \\ &= \left[1 + N^{-1} - \frac{\epsilon}{N}\right] \times \left[\frac{1}{bias}\left[2^{bias} - 2^{-bias}\right]\right] \\ &\approx \frac{2^{bias}}{bias} + \frac{2^{bias}}{N \times bias} - \frac{2^{-N} \times 2^{bias}}{N \times bias}. \end{aligned} \tag{61}$$

Clearly, the expected absolute error for a mantissa bit flip will be dominated by the exponent of the scalar.

We now analyze the expected value over the positive and negative exponent sets. The covariance is zero, therefore, $\mathbb{E}[X_{\alpha^+}X_\xi] = \mathbb{E}[X_{\alpha^+}]\mathbb{E}[X_\xi]$ and $\mathbb{E}[X_{\alpha^-}X_\xi] = \mathbb{E}[X_{\alpha^-}]\mathbb{E}[X_\xi]$. The expected value over the positive set of exponents is

$$\mathbb{E}[X_{\alpha^+}]\mathbb{E}[X_\xi] = \frac{2^{bias+1} - 1}{bias + 1} \times \left[1 + N^{-1} - \frac{\epsilon}{N}\right]. \tag{62}$$

The negative set of exponents yields an expected value

$$\mathbb{E}[X_{\alpha^-}]\mathbb{E}[X_\xi] = \frac{1 - 2^{1-bias}}{bias - 1} \times \left[1 + N^{-1} - \frac{\epsilon}{N}\right]. \tag{63}$$

Recognize that Eq. (63) is a decreasing function in both N and the $bias$. Recall that $bias \geq 3$, because $Z \geq 3$ as shown in Eq. (37), and $N > 0$. The expected absolute error given a bit flip in the mantissa and a negative exponent is bounded by

$$\mathbb{E}[X_{\alpha^-}]\mathbb{E}[X_\xi] \leq \frac{9}{16}, \tag{64}$$

which we show graphically in Figure 7. There are two important points: 1) The expected absolute error is less than one, and 2) the expected absolute error is dominated by the number of exponent bits. In Figure 7, eight exponent bits correspond to $bias = 127$, which is used in the single precision specification. Single precision has an expected absolute error of approximately 10^{-2}. Eleven exponent bits result in a bias value of 1023, which is used for IEEE-754 double precision. The expected absolute error for double precision is approximately 10^{-3}. Fifteen exponent bits are used for quad precision, which has a bias of 16383. The expected absolute error for a mantissa bit flip in quad precision is on the order of 10^{-5}.

Figure 7 also shows that the absolute error given a bit flip in the mantissa can be forced to be small. Combined with the relative error given a bit flip in the mantissa, as shown in Figure 6, the absolute and relative errors are fairly well behaved if the scalar is in the interval $(-1, 1)$, i.e., $\alpha \leq 2^{-1}$.

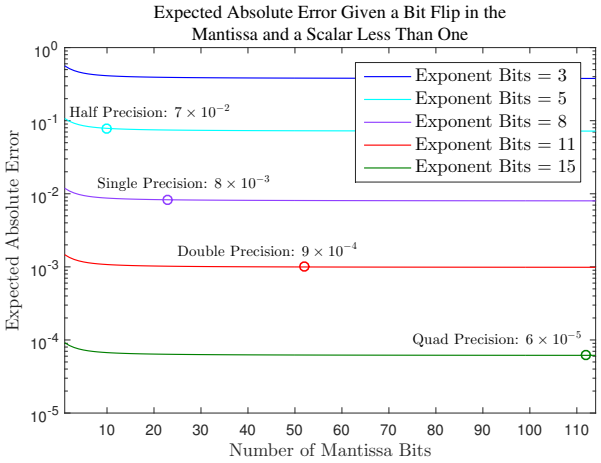


Figure 7: Expected absolute error given a bit flip in the mantissa and a scalar less than one. The absolute error is shown for the fewest number of exponent bits, as well as the number of exponent bits for half, single, double, and quad precision.

4.2.2. Exponent

The expected absolute error given a bit flip in the exponent is more tedious. We may algebraically isolate the error term, but the error term depends on the exponent bits. That is, $\alpha = 2^{e-bias}$, and $\tilde{\alpha} = 2^{\tilde{e}-bias} = 2^{e-bias\pm\eta}$. Precisely,

$$\tilde{\alpha} = 2^{-bias} \times 2^{(-1)^{bit_j}\times 2^j} \times \prod_{k=0}^{Z-1} 2^{bit_k \times 2^j}; \text{ for } j\text{-th bit flipped.}$$

The expected absolute error is

$$\frac{1}{ZK}\sum_{i=1}^{K}\sum_{j=0}^{Z-1}\left|2^{i-bias}\left(1-2^{(-1)^{bit_j}\times 2^j}\right)\right|. \tag{65}$$

Knowing whether the bit flipped (bit_j) is zero or one is difficult, which makes a closed form difficult to express. Equation (65) is computable, and we can identify a critical component of the summation.

First, recognize that for a biased exponent $e \leq bias$, (i.e., $\alpha \leq 2^0$), the most significant exponent bit is always zero.

Lemma 4.1 ($e \leq bias \iff exponent\_bit_{Z-1} = 0$)**.** *A biased exponent e is less than or equal to the bias, if and only if the most signficant exponent bit is* 0*.*

Proof 4.2 ($e \leq bias \iff exponent\_bit_{Z-1} = 0.$)**.** *Suppose* $e \leq bias$ *and* $exponent\_bit_{Z-1} = 1$*. The bias is defined to be* $bias = 2^{Z-1} - 1 = \sum_{i=0}^{Z-2} 2^i$*. e is defined to be the unsigned integer stored in the exponent bits, hence* $e = \sum_{i=0}^{Z-1} exponent\_bit_i \times 2^i$*. If* $exponent\_bit_{Z-1} = 1$*, then* $e = 2^{Z-1} + \sum_{i=0}^{Z-2} exponent\_bit_i \times 2^i > bias$*, which is a contradiction.*

Suppose $e > bias$ *and* $exponent\_bit_{Z-1} = 0$*.* $e = \sum_{i=0}^{Z-2} exponent\_bit_i \times 2^i + 0 \leq bias$*, which is a contradiction.* □

The property of biased exponents expressed in Lemma 4.1 implies that $bias - 1$ values in the summation will be $|a(1 - 2^{bias+1})|$, where $a \in (-1, 1)$. That is, for all exponents in the negative set, each exponent can always have the most significant bit flipped, creating an absolute error of $|a(1 - 2^{bias+1})|$, where $a \in (-1, 1)$.

The scalar's unperturbed exponent will negate this increase in the exponent to some extent. For example, a scalar $x = 2^{-(bias-1)}$ has the unsigned integer 1 stored in binary in its Z exponent bits. With the most significant bit flipped, the unsigned integer $(bias + 1) + 1$ is now stored in x's exponent bits. The $(bias + 1)$ term comes from flipping the most significant bit, and $+1$ is the value already present in x's lower-order exponent bits. The scalar x, now $\tilde{x}$, represents the floating point value $\tilde{x} = 2^2$. As we consider the scalars $x = 2^{-(bias-2)}, 2^{-(bias-3)}, \dots 2^{-1}$, the faulty scalars will be $\tilde{x} = 2^3, 2^4, \dots, 2^{bias}$. Given that the correct (unperturbed) scalars are all less than one, the absolute error between the correct and faulty scalars will always be larger than one given a bit flip in the most significant bit. Given that the number representable negative exponents $(bias - 1)$ is much smaller than 2^{bias}, the summation is dominated by the summing of $2^{bias} + 2^{bias-1} + \dots + 2^2$. This portion of the summation is from the most significant exponent bit flipping. There are $(Z - 1)$ lower-order exponent bits, which can never cause the faulty scalar to be larger than 2. That is, the worst the lower-order exponent flips can do is to create the binary pattern 2^0, which when combined with the largest possible mantissa value would create a scalar $\tilde{x} = 2 - \epsilon$.

For the positive set of exponents, the most significant bit will divide the number by large power of two, but the remaining bit flips will all produce a perturbed scalar that is larger than one. Conversely, the remaining bits of the exponent given a value $a \in (-1, 1)$, will produce a perturbed scalar that is less than two with probability $\frac{2Z-1}{2Z}$, which we show in Theorem 4.2. Because $a \in (-1, 1)$ and $\tilde{a} \in (-2, 2)$ and the sign cannot change because we assume the bit flip occurs in the exponent, then $|a - \tilde{a}| < 2$. That is, for the negative set of exponents, the expected absolute error is bounded by

$$\begin{aligned}
\mathbb{E}[error_{\text{abs}}]_{\alpha^-} &< \frac{2^{bias} - C + 2^{bias-1} - C + \dots + 2^2 - C + D \times (Z-1)(bias-1)}{Z(bias-1)} \\
&< \frac{\sum_{i=1}^{bias-1} 2^{i+1} - (bias-1) \times C + D \times (Z-1)(bias-1)}{Z(bias-1)} \\
&< \frac{\sum_{i=1}^{bias-1} 2^{i+1} + E \times Z(bias-1)}{Z(bias-1)} \\
&< \frac{2^{bias+1} - 4}{Z(bias-1)} + E, \qquad (66)
\end{aligned}$$

where $C = \max\{1.0, 1+\epsilon, 1+2\epsilon, \dots, 2-\epsilon\} = 2-\epsilon$. That is, C is the largest possible mantissa value. Because we consider the negative set of exponents, the largest (unscaled) mantissa also bounds the scalars $2^{-j} \times 1.\beta \leq C$, because the largest scalar obtainable given the negative exponent set is $|a_{\max}| = |2^{-1} \times (2 - \epsilon)| < 1.0$. We introduce a constant, D, which represents the largest possible absolute difference if the most significant exponent bit is not flipped. That is, $D = \max |a - \tilde{a}|$, which will always be less than two if the bit flipped is not the most significant, i.e., $D < 2$. The term $D \times (Z - 1)(bias - 1)$ accounts for the absolute error of the remaining $Z - 1$ exponent bits being flipped in the $bias - 1$ scalars, which always produce a faulty scalar strictly less than two, i.e., $|a - \tilde{a}| < 2$. Recognize that $D < 2$ and $C < 2$. To impose a strict (gross) upper bound, we may let $E = 2$, and substitute E for C and D, creating an over estimate of the expected absolute error given a bit flip in the exponent of scalars strictly less than one.

Theorem 4.2. *A bit flip in the exponent bits of an IEEE-754 scalar (a) that has a biased exponent $e \leq bias$ will produce a perturbed scalar $(\tilde{a})$ that is less than two, with a probability of $\frac{2Z-1}{2Z}$.*

Proof 4.3. *If the k-th bit is flipped $0 \to 1$ and $k \in 0, 1, 2, \dots, Z - 2$, then*

$$\tilde{e} = \left[\sum_{\substack{i=0 \\ i \neq k}}^{Z-2} exponent\_bit_i \times 2^i \right] + 2^k \leq \sum_{i=0}^{Z-2} 2^i.$$

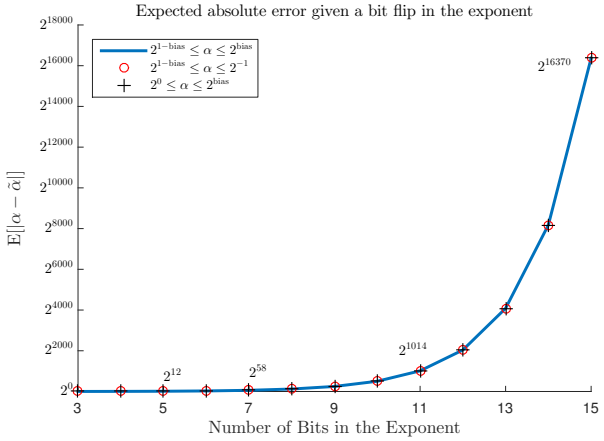


Figure 8: Expected absolute error given a bit flip in the exponent and a scalar less than one, greater than one, and over the full range. Half (2^{12}), Single (2^{56}), Double (2^{1014}), and Quad (2^{16170}) precision specifications are highlighted.

Table 6: Expected absolute error given a bit flip in exponent for scalar in the range $|a| \geq 1$ compared to the expected value of the positive set of exponents $\mathbb{E}[X_\alpha]$.

Spec.	$\mathbb{E}[X_\alpha]$	Approximate $\mathbb{E}[\lvert a - \tilde{a}\rvert]$; for $\lvert a\rvert \geq 1$.
Half Prec.	2^{12}	2^{12}
Single Prec.	2^{58}	2^{58}
Double Prec.	2^{1014}	2^{1014}
Quad Prec.	2^{16370}	2^{16370}

If the k-th bit is flipped $1 \to 0$ *and* $k \in 0, 1, 2, \ldots, Z-2$*, then*

$$\tilde{e} = \left[\sum_{\substack{i=0 \\ i \neq k}}^{Z-2} exponent\_bit_i \times 2^i \right] \leq \sum_{i=0}^{Z-2} 2^i.$$

By Lemma 4.1, the most significant bit is zero. Hence, if the bit flipped is $k = Z-1$*, it cannot flip* $1 \to 0$*.*

The probability of creating a faulty scalar with exponent $\tilde{e} \leq bias$ *is* $Z-1+Z-1+1 = 2Z-1$*. The total number of potential bit flips is* $2Z$*. Hence, if* $e \leq bias$*, then the resulting faulty scalar will have an exponent* $\tilde{\alpha} \leq 2^0$ *with probability* $\frac{2Z-1}{2Z}$*.*

We numerically evaluate the expected absolute error given a bit flip in the exponent in Figure 8. Recognize the positive and negative set give approximately the same expected value, because the most significant bit of values $a \in (-1, 1)$ forces the summation to include a large power of two. We also compare the the expected absolute error to that of the expected value of the exponent in Table 6. If all exponent bits are allowed to be faulty, then the expected absolute error behaves like the expected value of an exponential function.

4.2.3. Sign

A bit flip in the sign can be treated two ways, we may compute the expected value over the positive and negative sets, or we may treat a as constant, e.g., $\mathbb{E}[2a] = 2a$. Recognize that the negative set of exponents restricts the scalar such that $a \in (-1, 1)$ if α^-. If $a \in (-2, 2)$, then the sign bit will create an absolute error less than one, while if $|a| \geq 2$, then the absolute error will be larger than one.

4.3. Summary of Scalar Error Statistics

We summarize the statistics for each error measure derived in § 4.1 and § 4.2 in Tables 7 and 8. These are models for the correct and faulty components of the IEEE-754 representation when perturbed by a single

Table 7: Expected value of the relative error given a bit flip in a scalar ($\tilde{a}$).

Error Loc.	$error_{\text{rel}}$	Constraint	$\mathbb{E}[error_{\text{rel}}]$	Ref.
Mantissa	$\frac{1.\xi}{1.\beta}$	—	$\ln(2)\left[1+\frac{1}{N}\right]$	Eq. (55)
Exponent	$\lvert 2^{\pm\eta}-1\rvert$	—	$\frac{2^{bias}}{Z} < \mathbb{E} < \frac{2^{bias+1}}{Z}$	Eq. (56)
		positive η	$\frac{2^{bias+1}}{Z} < \mathbb{E} < \frac{2^{bias+2}-1}{Z}$	Eq. (58)
		negative η	< 1	Eq. (57)
Sign	2	—	2	Eq. (50)

bit flip. Consider the wide range of possible errors, e.g., the relative error from an exponent bit flip falls into two broad categories: very large and less than one. Similarly, the absolute error for a mantissa bit flip is dominated by the (non-faulty) exponent of the scalar. We consider an expected measure less than one to be "well behaved". This is a somewhat arbitrary definition, but the motivation stems from numerical analysis. If we can enforce that errors behave predictably, then we can devise schemes to dampen or maintain those errors, or possibly detect a subset of such errors. The choice of one also comes from observations in § 2. We have shown that the *expected* error tends to fall into two broad categories: larger than one, and less than one. We now extend these models to the multiplication operation, and the drawn higher-level conclusions about how the expected error models behave.

5. Model Multiplication Error Measures

Extending our analysis to multiplication is straight-forward, because only one operand can be faulty. We treat the non-faulty operand as a constant, and hence it operates as a scaling factor applied to our previous absolute error models and has no impact on the relative error. Consider two scalars, a and b. We now model the absolute and relative error measures for the operation $a \times b$.

5.1. Mantissa

For an error impacting the mantissa of one operand, the absolute error for multiplication is

$$\begin{aligned} error_{\text{abs}} &= |ab - \tilde{a}b| \\ &= |ab - b(a \pm \alpha \times 1.\xi)| \\ &= |ab - ab \pm b\alpha \times 1.\xi| \\ &= |b\alpha \times 1.\xi|, \end{aligned} \tag{67}$$

Table 8: Expected value of the absolute error given a bit flip in a scalar ($\tilde{a}$).

Error Loc.	$error_{\text{abs}}$	Constraint	$\mathbb{E}[error_{\text{abs}}]$	Ref.
Mantissa	$\lvert\alpha \times 1.\xi\rvert$	—	$\approx 2^{bias}$	Eq. (61)
Pos. α		$1 \leq \lvert a\rvert$	$\approx 2^{bias+1}$	Eq. (62)
Neg. α		$\lvert a\rvert < 1$	$\leq \frac{9}{16}$	Eq. (63)
Exponent	$\lvert a(1-2^{\pm\eta})\rvert$	—	$\approx 2^{bias+1}$	Eq. (35)
Pos. α		$1 \leq \lvert a\rvert$	$\approx 2^{bias+1}$	Eq. (35)
Neg. α		$\lvert a\rvert < 1$	$\approx 2^{bias+1}$	Eq. (66)
Sign	$\lvert 2a\rvert$		$\lvert 2a\rvert$	—
Pos. α and 2^{-1}		$1/2 \leq \lvert a\rvert$	> 1	—
Neg. α except 2^{-1}		$\lvert a\rvert < 1/2$	< 1	—

where $\tilde{a}$ has the form of Eq. (8). The relative error is

$$\begin{aligned} error_{\text{rel}} &= \frac{\lvert ab - \tilde{a}b\rvert}{\lvert ab\rvert} \\ &= \frac{\lvert b\alpha \times 1.\xi\rvert}{\lvert ab\rvert} \\ &= \frac{\lvert \alpha \times 1.\xi\rvert}{\lvert a\rvert} \\ &= \frac{\lvert \alpha \times 1.\xi\rvert}{\lvert \alpha \times 1.\beta\rvert} \\ &= \frac{1.\xi}{1.\beta}. \end{aligned} \tag{68}$$

5.2. Exponent

For an error impacting the exponent, $\tilde{a}$ takes the form of Eq. (13), resulting in an absolute error of

$$\begin{aligned} error_{\text{abs}} &= \lvert ab - \tilde{a}b\rvert \\ &= \lvert ab - ab \times 2^{\pm\eta}\rvert \\ &= \lvert ab(1-2^{\pm\eta})\rvert. \end{aligned} \tag{69}$$

The relative error is

$$\begin{aligned} error_{\text{rel}} &= \frac{\lvert ab - \tilde{a}b\rvert}{\lvert ab\rvert} \\ &= \frac{\lvert ab(1-2^{\pm\eta})\rvert}{\lvert ab\rvert} \\ &= \lvert 1-2^{\pm\eta}\rvert. \end{aligned} \tag{70}$$

Table 9: Error measures for faulty multiplication.

Location	Absolute Error	Ref	Relative Error	Ref
Mantissa	$\lvert b\alpha \times 1.\xi \rvert$	Eq. (67)	$\frac{1.\xi}{1.\beta}$	Eq. (68)
Exponent	$\lvert ab(1-2^{\pm\eta}) \rvert$	Eq. (69)	$\lvert 1-2^{\pm\eta} \rvert$	Eq. (70)
Sign	$\lvert 2ab \rvert$	Eq. (71)	2	Eq. (72)

5.3. Sign

For an error in the sign bit, $\tilde{a} = -a$. The absolute error is

$$\begin{aligned} error_{\text{abs}} &= |ab - \tilde{a}b| \\ &= |ab - b(-a)| \\ &= |ab + ab| \\ &= |2ab|, \end{aligned} \tag{71}$$

and the relative error is

$$\begin{aligned} error_{\text{rel}} &= \frac{|ab - \tilde{a}b|}{|ab|} \\ &= \frac{|2ab|}{|ab|} \\ &= 2. \end{aligned} \tag{72}$$

We summarize the error measures for multiplication in Tables 9.

5.4. Multiplication Expected Error

The expected relative error is the same as our scalar models, e.g., See Table 7. The absolute error is simply the scalar model's expected absolute error scaled by the non-faulty operand, which we consider to be b.

Table 10 summarizes the expected absolute error. Note that, we have not stated all possible conditions for $|b|$. It is possible that if $b = 1/error_{\text{abs}}$, then you could clearly have an absolute error less than one. We have stated the conditions for $|a|$ and $|b|$ such that we know when the expected value will behave as we have modeled.

6. Applications to Fault Tolerance

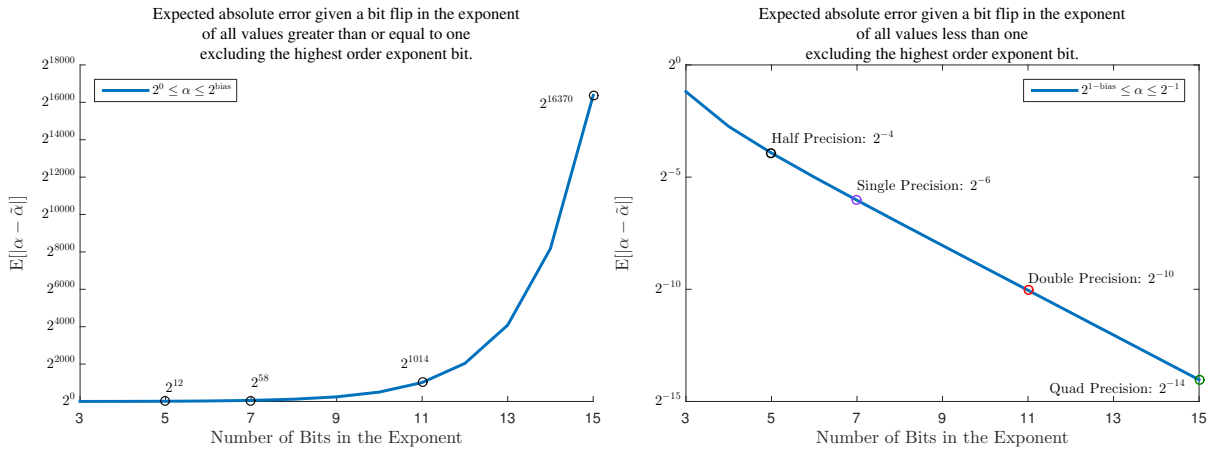
We have analyzed IEEE-754 scalars that experience a single bit flip, and have shown models for the errors introduced. A key point from the analysis is that the expected error, either absolute or relative, can fall into two very broad categories: greater than one and less than one. A common technique in our analysis has been to partition the exponents into positive and negative exponent sets (α^+, and α^- respectively). We now consider how these characteristics relate to injecting bit flips into scalars (or into scalar multiplication).

6.1. Constrained Exponent Bit Flips

In § 4.2.2, we showed that the most significant bit being flipped dominates the expected absolute error, e.g., see Theorem 4.2. We now consider the case that we are able to exclude the most significant bit from the expected value. We compute the expected absolute error excluding the most significant exponent bit for both the negative and positive set of exponents. Figure 9a plots the expected absolute error given a bit flip

Table 10: Expected value of the absolute error given a bit flip in scalar multiplication ($\tilde{a} \times b$).

Error Loc.	$error_{\text{abs}}$	Constraint	$\mathbb{E}[error_{\text{abs}}]$	Ref.
Mantissa	$\lvert b\alpha \times 1.\xi \rvert$	—	$\approx 2^{bias}$	Eq. (61)
Pos. α		$1 \leq \lvert b \rvert$ and $1 \leq \lvert a \rvert$	$\approx 2^{bias+1}$	Eq. (62)
Neg. α		$\lvert b \rvert < 1$ and $\lvert a \rvert < 1$	$\leq \frac{9}{16}$	Eq. (63)
Exponent	$\lvert ab(1-2^{\pm\eta}) \rvert$	—	$\approx 2^{bias+1}$	Eq. (35)
Pos. α		$1 \leq \lvert b \rvert$ and $1 \leq \lvert a \rvert$	$\approx 2^{bias+1}$	Eq. (35)
Neg. α		$\lvert b \rvert < 1$ and $\lvert a \rvert < 1$	$\approx 2^{bias+1}$	Eq. (66)
Sign	$\lvert 2ab \rvert$		$\lvert 2ab \rvert$	—
Pos. α and 2^{-1}		$1 \leq \lvert b \rvert$ and $1/2 \leq \lvert a \rvert$	> 1	—
Neg. α except 2^{-1}		$\lvert b \rvert < 1$ and $\lvert a \rvert < 1/2$	< 1	—



(a) Values greater than or equal to one.

(b) Values less than one.

Figure 9: Expected absolute error given a bit flip in the exponent, excluding the most significant bit for values less than one, and values greater than one.

Table 11: Probability that the relative error will be less than one given a bit flip in a scalar.

Specification	Mantissa	Exponent	Sign	$\Pr(error_{rel} < 1)$	$\Pr(error_{rel} \geq 1)$
Half Prec.	$\frac{10}{16}$	$\frac{1}{2} \times \frac{5}{16}$	$\frac{0}{16}$	0.78125	0.21875
Single Prec.	$\frac{23}{32}$	$\frac{1}{2} \times \frac{8}{32}$	$\frac{0}{32}$	0.84375	0.15625
Double Prec.	$\frac{52}{64}$	$\frac{1}{2} \times \frac{11}{64}$	$\frac{0}{64}$	0.89844	0.10156
Quad Prec.	$\frac{112}{128}$	$\frac{1}{2} \times \frac{15}{128}$	$\frac{0}{128}$	0.93359	0.06641

in the positive set of exponents, while excluding the most significant bit. We see the expected absolute error behaves like the expected value of the exponent, e.g., Eqs. (35) or (27).

Figure 9b plots the expected absolute error over the negative set of exponents, while excluding the most significant bit. The point is that if the most significant bit is excluded the expected absolute error is less than one. That is, we can remove the large error term from Eq. (66), yielding an expected absolute error that is strictly less than one when given a scalar less than one, e.g., Eq. (73).

$$\begin{aligned} \mathbb{E}[error_{\text{abs}}]_{\alpha^-} &< \overbrace{\frac{2^{bias+1}-4}{Z(bias-1)}}^{\text{Excluded}} + E \\ \mathbb{E}[error_{\text{abs}}]_{\alpha^-} &< 2. \end{aligned} \tag{73}$$

This property is particularly useful, as it shows that a bit flip in the exponent does not necessarily result in a large error, or even the expectation of a large error. The absolute error given a bit flip can be somewhat well-behaved if α can be constrained to the negative set of exponents.

Equation (73) and its original form, Eq. (66), present an upper bound on the expected absolute error. This is because E is an upper bound for the absolute error $|a - \tilde{a}| < 2$. If the scalars are constrained such that $a \in (-1, 1)$, the perturbed scalar is only order 2 if the bit flip creates the binary form of the bias. That is, the bit flip must create an exponent that corresponds to 2^0. The mantissa forces the bound to be less than 2, because the mantissa is bounded in the interval $[1, 2)$.

7. Overall Effect of Constrained Exponent Bit Flips

We have considered bit flips in specific locations of the representation. What matters is how these various effects interplay. A main observation is that bit flips in values less than one will produce absolute and relative errors less than one most of the time. The expected relative errors for multiplication, shown in Table 7, show that the mantissa is expected to have a relative error of approximately $\ln(2)$. An exponent bit flip can be both less than one and very large.

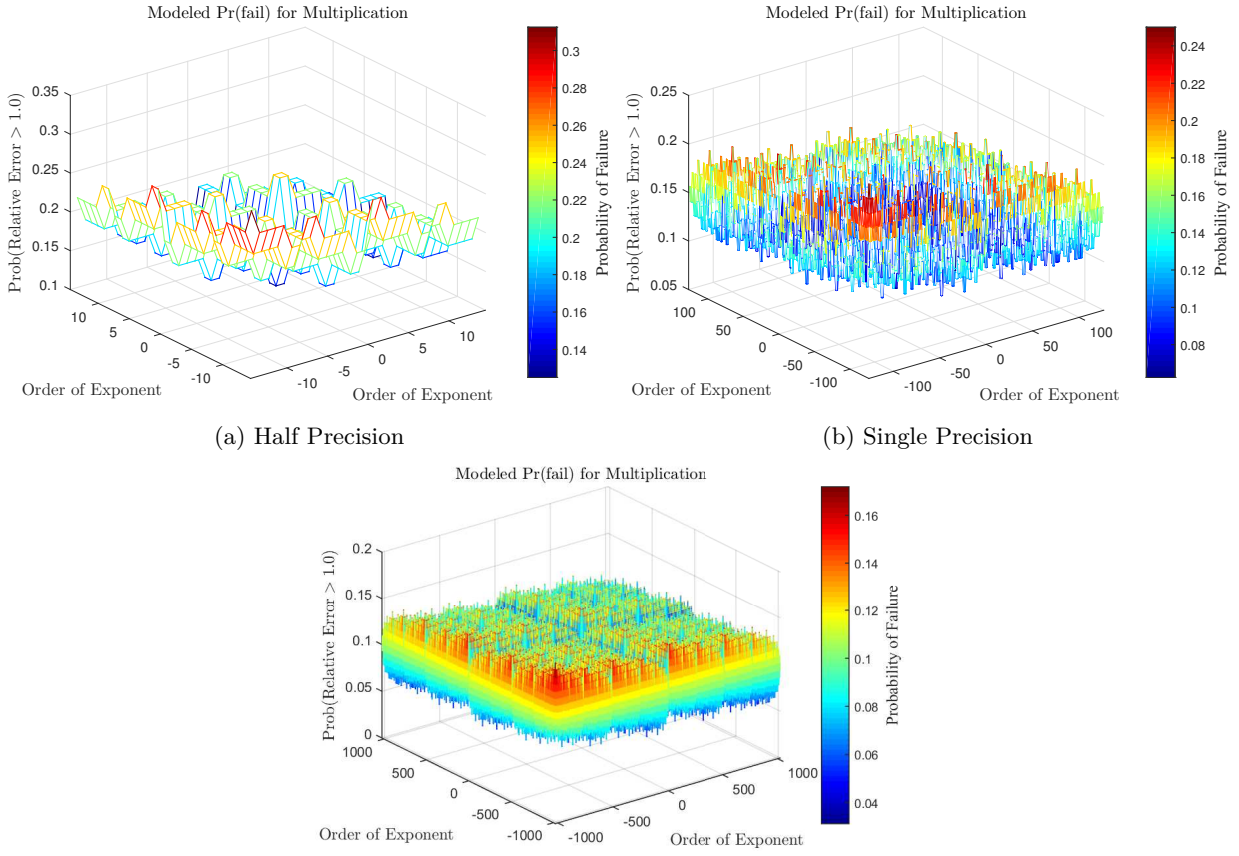
We consider an IEEE-754 scalar, which has $N_{bits} = Z + N + 1$, where Z is the number of exponent bits, N is the number of mantissa bits, and one sign bit. The probability that a bit will impact a specific region of the representation is then N/N_{bits} for a mantissa bit flip, Z/N_{bits} for an exponent bit flip, and $1/N_{bits}$ for a sign bit flip. Table 11 computes the probability that the relative error will be less than one, based on the expected value. The probability increases as the format increases from Half to Quad precision. This is because the mantissa bits dominate the expected value. Assuming bits are equally likely to be one or zero, then only half of the exponent bits will satisfy our condition of relative error less than one. A sign bit flip will always fail.

The values in Table 11 also model the expectation of the relative error for multiplication. We explore the behavior of the expected relative error by analyzing two scalars multiplied in half, single, and double precision, each with a fixed exponent, in Figure 10. Note that the probability of failure is the probability

that the expected error is larger than one. Figure 10a evaluates the possible relative errors for half precision. Observe that the center value of the colorbar is approximately 22%. Our model for half precision predicts 21.875% (see Table 11). If we compute the expected value across all scalars less than one, i.e., compute the expected value over the entire surface plot in the quadrant $[0, -14]$, we observe a probability of 0.21875, which agrees exactly with our model.

Figures 10b and 10c repeat our study using single and double precision. We observe that the relative error is larger than one approximately 16% of the time for single precision, and approximately 10% of the time for double precision. Computing the expected value over the scalars that are less than one, we observe the relative error larger than one 15.625% of the time for single precision, and 10.156% of the time for double precision. These values agree exactly with what our model predicts, as shown in Table 11.

This result is significant, because it shows that moving to quad precision is not necessarily bad. Naively, it would seem that the more bits used for a representation, the worse the errors should behave. The latter is not true, because the number of mantissa bits has increased significantly more than the number of exponent bits. Quad precision adds only 4 additional exponent bits compared to double precision, but adds 60 additional mantissa bits. The relative error of a bit flip in the mantissa is still quite large, $\ln(2)$, but this provides a clear direction for resilient algorithm design.



(a) Half Precision

(b) Single Precision

(c) Double Precision

Figure 10: Probability that the expected relative error is larger than one, given a single bit flip in scalar multiplication for: (a) half, (b) single, and (c) double precision.

7.1. Expected Absolute Error Given a Scalar

We now compute the probability that the absolute error will be less than one given a scalar less than one. If the scalars are strictly less than one, then the biased exponent will be in the range $[1, 1022]$. We must exclude the bit pattern of 1023, because scalars with exponents 2^0 are strictly less than two. Table 10 summarized the expected absolute error for a bit flip in scalar multiplication. We showed in Eq. (73) that the expected absolute error for scalars less than one, is dominated by the most significant exponent bit. For scalars less than one, $a \in (-1, 1)$, the mantissa and sign can never introduce an absolute error larger than one. Of all exponent bit flips, the majority will not create an absolute error larger than one. Specifically, there are $bias - 2$ exponents representable for scalars less than one. Because we constrain the scalars to be less than one, but must count the excluded biased exponent 1023, the total number of possible bit flips is

$$Total_{bitflips} = (bias - 1) \times (N + Z + 1).$$

The probability of observing an absolute error less than one is then

$$\frac{\overbrace{(bias-2)\times N}^{Mantissa} + \overbrace{(bias-2)\times 1}^{Sign} + \overbrace{(bias-2)\times(Z-1)}^{Exponent} - \overbrace{(Z-1)}^{Excluded}}{(bias-1)\times(N+Z+1)}. \tag{74}$$

Note that the "Excluded" term removes the $Z - 1$ possible bit flips that would create the excluded exponent 2^0 (i.e., the bias). Because we consider a single bit flip, there are exactly $Z - 1$ bit flips that can create the binary pattern corresponding to 2^0. These excluded bit flips all take the form of flipping a zero to a one, to create the binary pattern $0111 \ldots 1$. That is, we must not count the exponent bit flips that would create the the bias value in the exponent storage. If we allowed these, the absolute error would be bounded by 2.

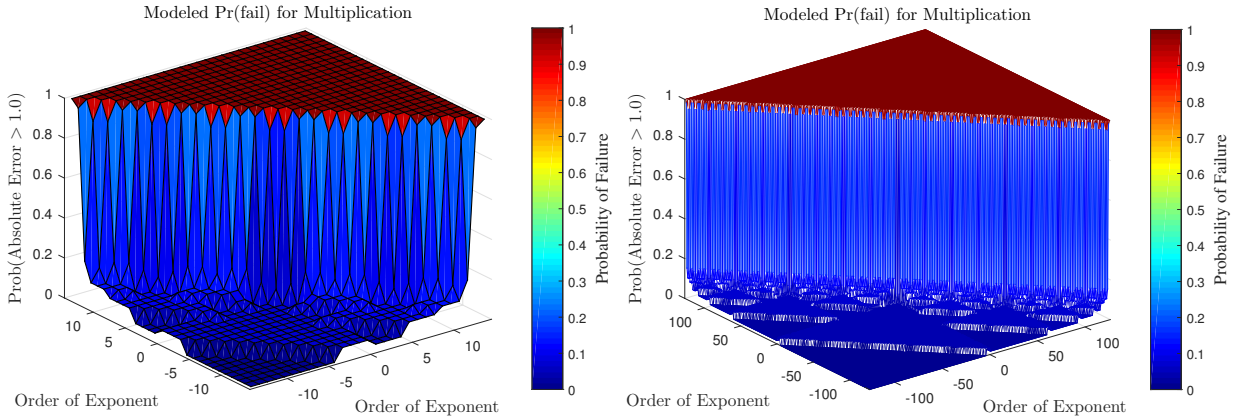
7.2. Expected Absolute Error Given Scalar Multiplication

We compute the probability that the absolute error will be less than one given a bit flip in scalar multiplication. Unlike the relative error, both scalars impact the absolute error. Figure 11 visualizes the probability that the absolute error will be larger than one for the half, single and double precision formats. We construct each figure by fixing the exponent of each scalar and using the expected value of the mantissa. In [9], similar figures are constructed from Monte Carlo trials to approximate the expected value of the mantissa. Elliott et al. [9, Figure 2] used dot products rather than scalars. Notice the structure of our plots are very similar. The region showing the probability when both scalars are less than one is flat and near zero. The difference is the gradient that transitions from low probability to high. This is expected, because outside of the region where both scalars are less than one, the absolute error can be both small and large. Specifically, mantissa bit flips can "fail", e.g., see Eq. (67).

To emphasize the impact of the mantissa, Figure 12 repeats our experiment and fixes the mantissa errors such that we observe the largest possible mantissa error for all mantissa bit flips. That is, we always flip the most significant mantissa bit, rather than show the average. Clearly, this is not realistic. We have shown the analytic form of the expected mantissa error in Eq. (22), which is much smaller than the largest mantissa error (1.5). The effect of modeling all worst-case mantissa errors is that the mantissa fails more often. This effect is most easily visualized by the reduction in the "drip" effect seen in Figure 11a, but the impact is essential in understanding how the low probability region behaves.

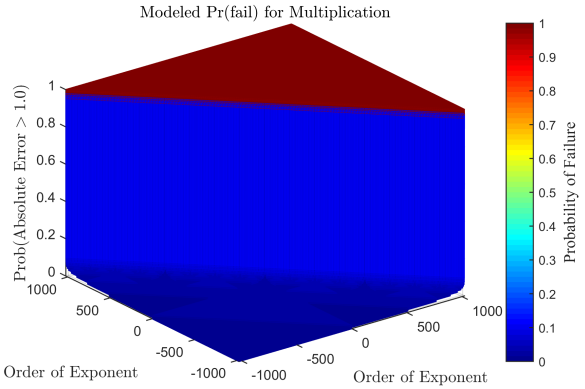
We explore the region of scalars strictly less than one in Figure 13 for the half precision format. Figures 13a and 13b perform the experiment using the expected value of the mantissa error, and Figures 13c and 13d use the worst-case mantissa error. The right-most plots avoid the interpolation that the surface plot introduces and provide a clearer picture of which combination of scalars are producing absolute errors larger than one. The difference between the two experiments is the increased failure of scalars with exponents of 2^{-2} and 2^{-1}. This maps to the two extra orange squares in Figure 13d. We show a zoomed-in view for the single and double precision formats in Figures 14a and 14b, respectively. Recognize the spikes in each. The spikes indicate a higher probability of observing an absolute error larger than one. Due to lack of fidelity with single and double precision, we use half precision in latter examples.

The structure in Figure 13 is the result of two side effects introduced from multiplication:



(a) Half Precision

(b) Single Precision



(c) Double Precision

Figure 11: Probability that the expected absolute error is larger than one, given a single bit flip in scalar multiplication for: (a) half, (b) single, and (c) double precision.

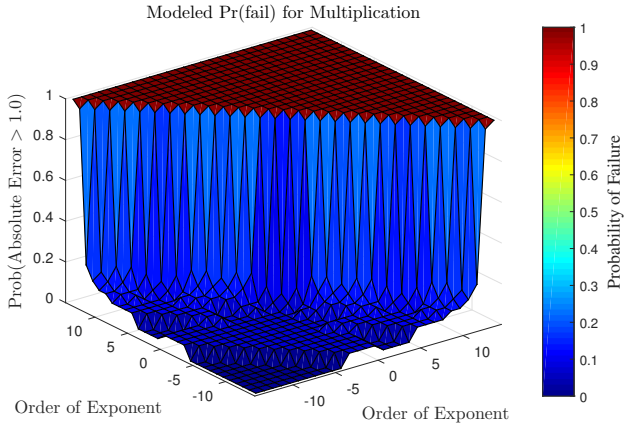
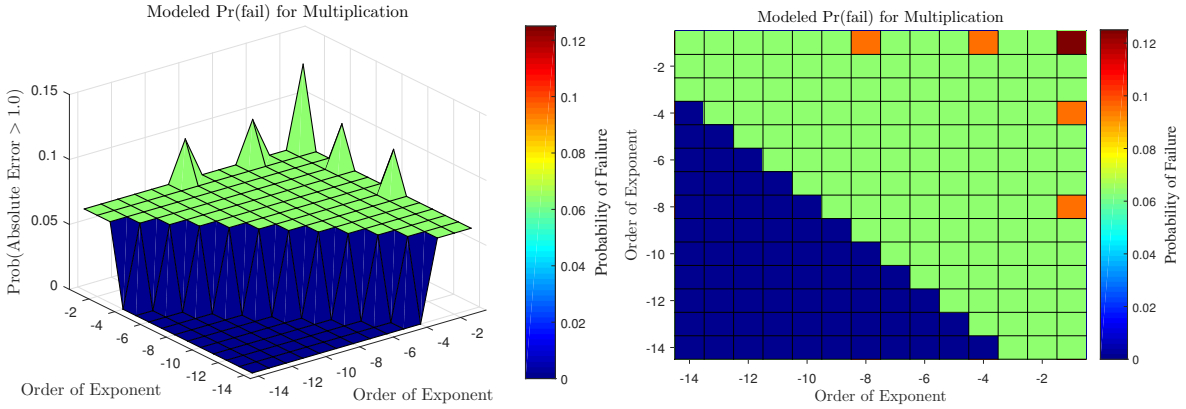
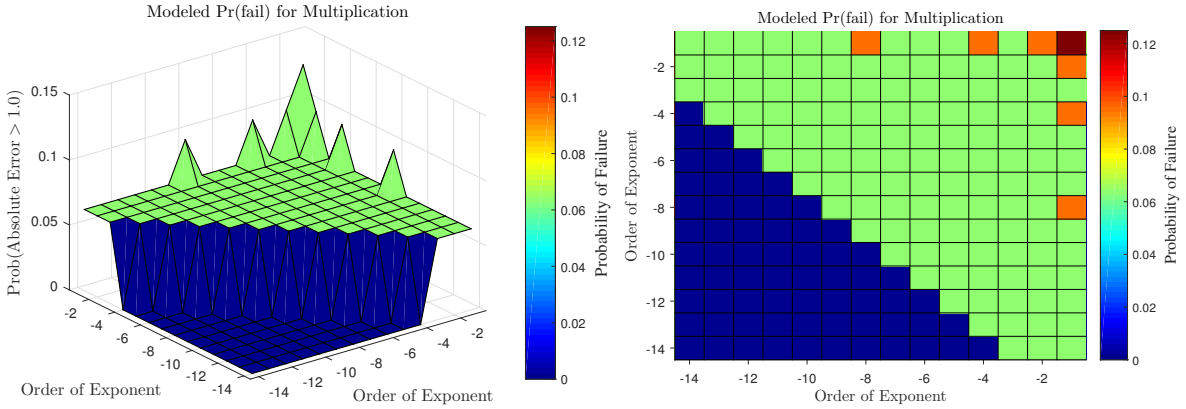


Figure 12: Absolute error experiment for half precision using the worst-case mantissa error for all mantissa errors.



(a) Half precision surface using the expected mantissa error.

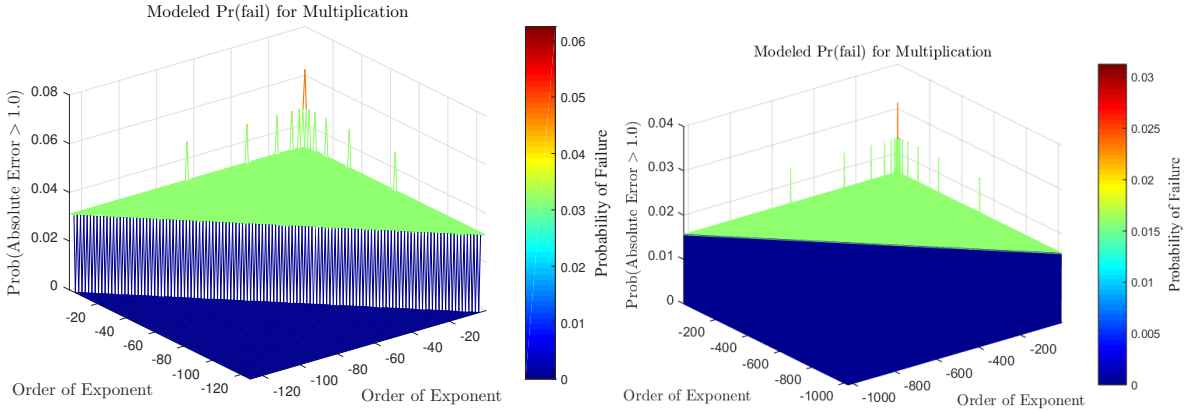
(b) Half precision overhead view using the expected mantissa error.



(c) Half precision surface using the worst-case mantissa error.

(d) Half precision overhead view using the worst-case mantissa error.

Figure 13: Probability that the expected absolute error is larger than one, given a single bit flip in scalar multiplication for half precision and scalars in the range $[0, 1)$.



(a) Single precision surface using the worst-case mantissa error.

(b) Double precision surface using the worst-case mantissa error.

Figure 14: Probability that the expected absolute error is larger than one, given scalars in the range $[0, 1)$ and a single bit flip in scalar multiplication for (a) single and (b) double precision.

- First, the green spikes map to the excluded bits from Eq. (74). Given scalars in the range of $[0, 1)$ there are $Z - 1$ bit flips that create the exponent 2^0, i.e., the bias value. We must exclude the exponent 2^0, because it enables the mantissa bits to fail.
- Second, the non-faulty scalar functions as a "scaling factor". There are cases where the operands are sufficiently small that they annihilate the large error produced by the most significant bit flipping.

We now address the effect of each case.

7.2.1. Excluded Bits

To ensure the absolute error remains less than one, we require that the scalars operated on be strictly less than one, e.g., see Eq. (74). If we require that both scalars have exponent values $\alpha \leq 2^{-1}$, then the $Z - 1$ bit flips that will create the exponent 2^0 are only impactful if the non-faulty operand has exponent 2^{-1}.

Lemma 7.1. *Given two scalars with exponents in the interval $[2^{-(bias-1)}, 2^{-1})$, it is impossible for a bit flip in an exponent bit, bit $\in \{exponent\_bit_{Z-2}, exponent\_bit_{Z-3}, \ldots, exponent\_bit_0\}$, to create a absolute error larger than one if the non-faulty operand has exponent $\alpha \leq 2^{-2}$.*

Proof 7.1. *Let two scalars be $(a, b) \in (-1, 1)$. Suppose a bit flip perturbs a such that it creates the bias pattern in the exponent, i.e., $\tilde{\alpha}_a = 2^0$. Suppose the exponent for the non-faulty scalar b is less than 2^{-1}. The resulting perturbed exponent will be $\tilde{a} \times b = 2^0 \times 1.\beta_a \times 2^{-2} \times 1.\beta_b = 2^{-2} \times 1.\beta_a \times 1.\beta_b$.*

Any mantissa, $1.\beta$, is bounded above by $1.\beta < 2$. Hence, the faulty multiplication is bounded by $|\tilde{a} \times b| < 2^{-2} \times 4$, and the absolute error cannot be larger than one, because $|a \times b| < 1$. □

The spikes in Figures 13c, 14a, and 14b occur when the non-faulty scalar has exponent 2^{-1}. This effect is symmetric because multiplication is commutative. Recognize the number of spikes in Figure 13c is $Z - 1 = 5 - 1 = 4$ along either boundary, because the boundary represents a scalar with exponent 2^{-1}.

7.2.2. Most Significant Bit Flip Mitigated

Next, we explain the two-level structure seen in Figure 13c. Given scalars in the range $2^{bias-1}, 2^{-1}$, there are $bias - 1$ such scalars, and therefore $bias - 1$ most significant exponent bits that can be flipped leading to a large absolute error. Given scalar multiplication, it is possible that if both scalars are sufficiently small, their product will produce an exponent sufficient to cancel the large multiplicative error. Given scalars in the range $[0, 1)$, a bit flip in the most significant exponent bit of an operand in scalar multiplication will produce an absolute error less than one $(bias - 4) \times (bias - 3)$ number of times.

Lemma 7.2. *Given two scalars with exponents in the interval $[2^{-(bias-1)}, 2^{-1})$ the most significant bit flip in either operand of scalar multiplication will produce an absolute error less than one $(bias - 4) \times (bias - 3)$ number of times.*

Proof 7.2. *Let two scalars be $(a, b) \in (-1, 1)$. Suppose a bit flip perturbs a such that it creates the bias pattern in the exponent, i.e., $\tilde{\alpha} = 2^0$. Suppose the most significant exponent bit is flipped in either operand of scalar multiplication. The multiplicative error introduced by the most significant exponent bit flip is 2^{bias+1}. To ensure the absolute error is less than one, the resulting complete exponent (α) from the faulty product must be $\alpha \leq 2^{-2}$, otherwise the mantissa values can cause failure.*

To have $\alpha \leq 2^{-2}$, the sum of the exponents must satisfy $exp_a + exp_b + bias + 1 \leq -2$. That is, the exponents of the operands must be sufficiently small to cancel the large multiplicative error 2^{bias+1} and remain small enough to ensure that the mantissa bits cannot produce a large absolute error.

The smallest exponent representable is $2^{-(bias-1)}$. Suppose $exp_b = -(bias - 1)$. This yields a bound of $exp_a - (bias - 1) + bias + 1 \leq -2$, $exp_a \leq -4$ if b has an exponent of $2^{-(bias-1)}$. If b becomes larger, than a must become smaller. The number of products satisfying this constraint is the triangle sum of $\frac{(bias-4)\times(bias-3)}{2}$. Repeating this analysis considering a as the smallest value, and b as the largest yields $2 \times \frac{(bias-4)\times(bias-3)}{2}$, or $(bias - 4) \times (bias - 3)$ number of products will cancel the most significant bit flip while ensuring the mantissa cannot force the error to be larger than one. □

Table 12: Probability that the absolute error will be larger than one given a bit flip in scalar multiplication with $(a, b) \in (-1, 1)$.

Specification	$\Pr(error_{abs} \geq 1)$
Half Prec.	0.04273
Single Prec.	0.01625
Double Prec.	0.00785
Quad Prec.	0.00391

The reasoning used in Proof 7.2 is clear when observing Figure 13. To compute the number of products that will negate the most significant bit flipping, we only need to sum the blue triangle in Figure 13d. For half precision, the bias is 15, and $bias - 4 = 12$. The blue triangular region represents the scalar products that satisfy $exp_a + exp_b + bias + 1 \leq -2$. That is, $-4 + (-14) + 15 + 1 = -2 \leq -2$.

7.3. Probability of an Absolute Error Larger Than One

The prior subsections have addressed two effects that multiplication has on the absolute error created should a bit flip in either operand. We now unify the concepts presented to compute the probability that the absolute error will be greater than or equal to one given scalars $(a, b) \in (-1, 1)$ and scalar multiplication.

First, the number of exponents possible given scalars in the interval $(-1, 1)$ is $(bias - 1)$. Multiplication has two operands each having $N + Z + 1$ bits. Therefore, the total number of bits that can be flipped are

$$Total\_bits = 2 \times (bias - 1)^2 \times (N + Z + 1). \tag{75}$$

The bit flips that can create the bias were addressed in Lemma 7.1. These "excluded" bit flips can only be impactful if the one of the scalars has exponent 2^{-1}, and there are only $Z - 1$ such bit flips given all representable exponents in the range $[0, 1)$. The total ways to have a bit flip create 2^0 is

$$Bit\_flips\_to\_bias = 2 \times (Z - 1). \tag{76}$$

The most significant exponent bit flip is not impactful for $(bias - 4) \times (bias - 3)$ of the total bit flips.

$$bad\_significant\_exp\_bit\_flips = 2 \times (bias - 1)^2 - (bias - 4) \times (bias - 3). \tag{77}$$

The probability that the absolute error will be larger than one is

$$\Pr(error_{abs} > 1) = \frac{Bit\_flips\_to\_bias + bad\_significant\_exp\_bit\_flips}{Total\_bits}. \tag{78}$$

We evaluate Eq. (78) for the formats half, single, double, and quad precision in Table 12. We also compared our modeled probability against what we observed in our experiments using the worst-case mantissa error, and obtained a perfect fit. That is, the difference between our modeled probability and what we observed is zero. Our model represents an upper bound because our proofs and derivations are designed to address the largest possible mantissa values, i.e., $1.\beta = 2 - \epsilon$. This difference is visuallized in Figure 13. Figure 13a has fewer spikes than Figure 13c. This is because the product of the expected mantissa error is $1 + 1/N$, which is less than 1.5 (the worst-case).

8. Application of Constrained Exponent Bit Flips

It is not immediately obvious how we could simply remove a bit from consideration in the expected value. Recognize that restricting values to be less than one is the same effect that normalizing a vector or

equilibrating a matrix achieves. The concept of operating on values in the interval $(-1, 1)$ is very common in numerical mathematics. For example, the Arnoldi process [2], which forms the basis for many Krylov subspace linear solvers and eigensolvers constructs an orthonormal set of basis vectors. Elliott et al. [9] instrumented the GMRES solver, and showed that the probabilities we have shown, hold inside the Arnoldi process.

This is a strong result. The errors from a bit flip seem extremely unpredictable, yet if data is scaled, then the errors are "well behaved" most of the time. Clearly, our definition of well behaved is arbitrary, but there is strong evidence that bounding errors is an effective fault tolerance mechanism for iterative linear solvers, e.g., see Elliott et al. [8].

Another implication of this work stems from § 7.2.2. Lemma 7.2 established that as values become smaller, the expected absolute error goes to zero. That is, given sufficiently small scalars, the product is sufficiently small to cancel the large error introduced. The relative error will always have a 50/50 chance of being large or small, but the absolute error will be less than one the majority of the time. This is important for algorithms that generate normalized values, as these values may becomes very small, but the absolute error will always be less than one.

9. Conclusion

We have used analytic modeling and statistical analysis to show how a bit flip in the representation of an IEEE-754 floating point behaves. We have computed the expected absolute and relative error, and shown that these measures fall into two broad categories: less than one, and very large. We have rigorously justified the observations in related work, and shown that the findings in [9] hold for formats besides binary64. We have shown that the number of bits is only marginally important; rather the scaling of the values is more important when it comes to the errors introduced from a bit flip.

References

[1] B. Adamczewski. The many faces of the Kempner number. *J. Integer Seq.*, 16(2):34, 2013.

[2] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[3] P. G. Bridges, K. B. Ferreira, M. A. Heroux, and M. Hoemmen. Fault-tolerant linear solvers via selective reliability. *ArXiv e-prints*, June 2012. Provided by the SAO/NASA Astrophysics Data System.

[4] G. Bronevetsky and B. de Supinski. Soft error vulnerability of iterative linear algebra methods. In *Proceedings of the 22nd Annual International Conference on Supercomputing*, ICS '08, pages 155–164, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-158-3. doi: 10.1145/1375527.1375552.

[5] Marc Casas, Bronis R. de Supinski, Greg Bronevetsky, and Martin Schulz. Fault resilience of the algebraic multi-grid solver. In *Proceedings of the 26th ACM International Conference on Supercomputing*, ICS '12, pages 91–100, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1316-2. doi: 10.1145/2304576.2304590. URL `http://doi.acm.org/10.1145/2304576.2304590`.

[6] T. Davies and Z. Chen. Correcting soft errors online in LU factorization. In *Proceedings of the 22nd International Symposium on High-Performance Parallel and Distributed Computing*, pages 167–178, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1910-2. doi: 10.1145/2462902.2462920.

[7] T. Davies, C. Karlsson, H. Liu, C. Ding, and Z. Chen. High performance LINPACK benchmark: A fault tolerant implementation without checkpointing. In *Proceedings of the 25th Annual International Conference on Supercomputing*, pages 162–171, May 2011.

[8] James Elliott, Mark Hoemmen, and Frank Mueller. Evaluating the impact of SDC on the GMRES iterative solver. In *28th IEEE International Parallel & Distributed Processing Symposium*, Phoenix, USA, May 2014.

[9] James Elliott, Mark Hoemmen, and Frank Mueller. Exploiting data representation for fault tolerance. In *Proceedings of the 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, pages 9–16, 2014. ISBN 978-1-4799-7562-4. doi: 10.1109/ScalA.2014.5. URL `http://dx.doi.org/10.1109/ScalA.2014.5`.

[10] N. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002. doi: 10.1137/1.9780898718027. URL `http://epubs.siam.org/doi/abs/10.1137/1.9780898718027`.

[11] Kuang-Hua Huang and Jacob A. Abraham. Algorithm-based fault tolerance for matrix operations. *IEEE Trans. Comput.*, C-33(6):518–528, June 1984.

[12] Piyush Sao and Richard Vuduc. Self-stabilizing iterative solvers. In *Proceedings of the Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, pages 4:1–4:8, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2508-0. doi: 10.1145/2530268.2530272.

[13] M. Shantharam, S. Srinivasmurthy, and P. Raghavan. Characterizing the impact of soft errors on iterative methods in scientific computing. In *Proceedings of the 25th International Conference on Supercomputing*, ICS '11, pages 152–161, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0102-2. doi: 10.1145/1995896.1995922.

[14] M. Shantharam, S. Srinivasmurthy, and P. Raghavan. Fault tolerant preconditioned conjugate gradient for sparse linear system solution. In *Proceedings of the 26th ACM International Conference on Supercomputing*, ICS '12, pages 69–78, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1316-2. doi: 10.1145/2304576.2304588.

[15] Panruo Wu and Zizhong Chen. FT-ScaLAPACK: Correcting soft errors on-line for ScaLAPACK Cholesky, QR, and LU factorization routines. In *Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing*, pages 49–60, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2749-7. doi: 10.1145/2600212.2600232. URL `http://doi.acm.org/10.1145/2600212.2600232`.