Probabilistic Interval-Valued Computation: Toward a Practical Surrogate for Statistics Inside CAD Tools

Amith Singhee, Claire F. Fang, James D. Ma, Rob A. Rutenbar
Dept. of ECE, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213 USA
{asinghee,jdma,rutenbar}@ece.cmu.edu, claire_fang@alumni.cmu.edu

Abstract

Interval methods offer a general, fine-grain strategy for modeling correlated range uncertainties in numerical algorithms. We present a new, improved interval algebra that extends the classical affine form to a more rigorous statistical foundation. Range uncertainties now take the form of confidence intervals. In place of pessimistic interval bounds, we minimize the probability of numerical "escape"; this can tighten interval bounds by 10X, while yielding 10-100X speedups over Monte Carlo. The formulation relies on three critical ideas: liberating the affine model from the assumption of symmetric intervals; a unifying optimization formulation; and a concrete probabilistic model. We refer to these as *probabilistic intervals*, for brevity. Our goal is to understand where we might use these as a surrogate for expensive, explicit statistical computations. Results from sparse matrices and graph delay algorithms demonstrate the utility of the approach, and the remaining challenges.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids

General Terms

Algorithms, Design, Statistics

Keywords

Intervals, DFM, algorithms

1. Introduction

The scaling of technologies toward the nanometer regime brings with it a challenging increase in the amount of variability we must model, manage, and optimize, across all phases of design. Variation sources may be global (e.g., wafer-level process problems) or local (e.g., random dopant variation in a single device), and posses a complex spatial or temporal correlation structure. These problems have generated a wave of new statistically "aware" tools and methods, e.g., [1]-[4].

It is worth noting, however, that the most successful of these techniques are application-specific, and depend for their success on a few inflexible statistical attributes of the problem at hand. Statistical static timing is an excellent example, enabled by the facts that delays are additive, and delay statistics are well modeled as normal distributions [1],[2]. Minimization of power under timing yield exploits the fact that a chance-constrained model of the problem yields a deterministic 2nd-order cone program [3]. Connections between time-domain circuit moments and moments of probability distributions enable a similarly attractive variational analysis for linear circuits [4].

Unfortunately, not every variational problem we seek to solve has a tractable analytical form. *What then*? Monte Carlo analysis remains the gold standard for "arbitrary" problems -- accurate, but often intractably slow. Are there other, general options?

Another alternative, with a surprisingly long history, is *interval-valued* analysis [5]. The key idea is to replace individual real values, such as x=3, with *finite ranges* on the real line, such as x=[1,4], and construct a suitable algebra of operators that supports interval-valued arithmetic and basic nonlinear functions such as *exp*( ) and *log*( ). Ideally, we can take a conventional numerical algorithm, and replace it operator by operator with an interval-valued version. Of course, this is not intrinsically a statistical model, but rather, a model of the uncertainty in the *extent* of these ranges. One must transform the distribution statistics of the problem into some suitable set of *range uncertainties*, with some concomitant loss of fidelity. See [7] for an early application in the domain of interconnect modeling.

The problem with these "classical" intervals is that, without any explicit mechanisms to track essential correlations among interval operands, range estimation errors explode during complex calculations. In the mid 1990s, de Figueiredo and Stolfi suggested the first workable attack on the correlation problem, the so-called *affine interval* form [6]. Each affine intervals comprises a nominal value with a set of shared uncertainty terms, which represent symmetric excursions about the nominal. Different operands can share elements among the set of uncertainty terms, thus preserving some correlation data. See [8] for an example of using these ideas in analog optimization.

Unfortunately, the affine form alone is still insufficient to model arbitrary variational CAD problems. The essential problem, first addressed in [9], is that the affine form is a purely deterministic model, and strives always for the range uncertainties it represents to enclose 100% of all possible numerical outcomes (from Monte Carlo simulation). This is deeply problematic if the scenarios we seek to model are comprised of random perturbations with unbounded support--true for any Gaussian pdf--or if the variations we seek to approximate as finite intervals are long-tailed pdfs, with negligible probability mass at their tails. There have been several heuristic attempts to create a workable statistical interpretation of the affine interval form, e.g., [10],[11] use affine computations, but interpret the resulting affine form as a combination of random variables, for interconnect model reduction.

In this paper, we present a more rigorous interval algebra that rebuilds the affine form on a practical statistical foundation. Range uncertainties now take the form of confidence intervals (extending ideas from [9]). In place of impractically pessimistic interval bounds, we minimize the probability of numerical "escape" to something work-

DAC 2006, July 24–28, 2006, San Francisco, California, USA.
Copyright 2006 ACM 1-59593-381-6/06/0007…$5.00

ably small: this can produce interval bounds that are tighter by 10X, while still faster than Monte Carlo by 10-100X. The formulation relies on three critical ideas: liberating the affine model from the assumption of symmetric intervals; a unifying optimization formulation; and a probabilistic model that exploits classical confidence interval concepts. Note that we are still modeling finite range uncertainties: the method uses statistical ideas to tighten the bounds, but does not propagate any explicit pdf through chains of operators. We must still map, in some empirical fashion, between pdfs and intervals, to use these ideas. We refer to the resulting method as *probabilistic intervals*, for brevity. Our goal is to understand where we may be able to use these intervals as a surrogate for expensive, explicit statistical computations inside numerical algorithms.

The paper is organized as follows. Sec. 2 gives a brief review of the classical affine form, and enumerates some of its shortcomings. Sec. 3 gives a method to support an asymmetric interval form. Sec. 4 discusses the *minvolume approximation*, a method for tighter intervals bounding on binary operators. Sec. 5 develops confidence interval concepts for probabilistic bounding. Sec. 6 presents numerical results from representative matrix and graph problems. Sec. 7 offers concluding remarks.

2. The Classical Affine Interval Form

The affine interval form [6] comprises a central value and a set of additive, symmetric excursions:

$$\hat{x} = x_0 + \sum_{i=1}^{N} x_i \varepsilon_i \qquad (1)$$

where each ε_i takes values in [-1,1] and is referred to interchangeably as an *error symbol* or an *uncertainty term*. Different variables can share error symbols, thus modeling first-order correlations. A pair of affine intervals $\hat{x}, \hat{y}$ with shared errors symbols defines a center-symmetric polytope, which we denote as U^{xy} (Fig. 1a); U^{xy} often defines a much smaller feasible range than that of the simple bounding rectangle.

Addition, subtraction and simple scaling are easily seen to yield the affine form directly. The general rule for approximating the result of a non-affine operation (e.g., $\times, \div, exp(\ )$) on affine operands is to seek an affine form that is a linear combination of the operands along with a new term (ζ) to account for the error. The range of the actual result should lie within the range of this affine approximation. Consider a unary operation like $\hat{y} = log(\hat{x})$ (Fig. 1b). As the N+1 coefficients of y (ε_i and ζ) vary, they define a pair of secants that bound the nonlinearity. [6] in general chooses coefficients to guarantee perfect conservatism in the final result.

The affine form is both elegant and attractive--yet, in our experience it has three significant weaknesses:

- **Only symmetric variations:** the affine form is a centered one, with symmetric variations about the nominal. This is surprisingly limiting, especially for strongly nonlinear operators. This can lead to significant undershoot/overshoot in the computed range (Fig. 1b), or in the worst case, sign errors.
- **Weak operator bounds:** [6] chooses coefficients that are easy to calculate, and often yield overly pessimistic bounds. We would prefer trade-offs that favor tighter bounds, at the expense of computation cost, and some numerical "escape".
- **No statistical foundation:** although it is tempting to interpret the ε_i as uniform random variables on [-1,1], in fact the core theory of the affine form is purely deterministic, focused only on the calculation of range bounds.

Our probabilistic interval formulation strives to repair all three of these problems. We describe how, in the following sections.

3. Allowing Asymmetry: Enforced Bounds

To solve the overshoot/undershoot problem (see again Fig. 1b), we enhance the basic affine interval with two *enforced bounds*. These are different from the symmetric bounds implied when the error symbols reach their most extreme values. The idea is simple: if we know from the form of the nonlinearity which ranges of values $\hat{x}$ *cannot* possibly take, we add this information to the interval model, thusly:

$$\{\hat{x}, [x]\} = \left\{x_0 + \sum_{i=1}^{N} x_i \varepsilon_i, x_l, x_h\right\} \qquad (2)$$

Here, x_l, x_h are the lower and upper enforced bounds, respectively. Hence, $\hat{x}$ cannot take any value outside $[x_l, x_h]$ ($[x]$ in short).

Now, the questions is how to *use* this information. Performing operations on such intervals entails two steps: (i) Compute the *symmetric* interval of the result, $\hat{z} \leftarrow \{\hat{x}, [x]\}$. (ii) Compute the *bounds* on the result, $[z] \leftarrow \{\hat{x}, [x]\}$. Fig. 2a shows how overshoot is eliminated for the *log*( ) function. Fig 2b shows the effect of asymmetric enforced bounds on the joint range of two affine intervals: the polytope is now truncated. This will add computational complexity when we try to tighten the bounds for binary operations, which we address next.

4. Tighter Bounds: the Minvolume Approximation

The general rule for approximating the result of a non-affine operation is to seek an affine form that is a linear combination of the op-

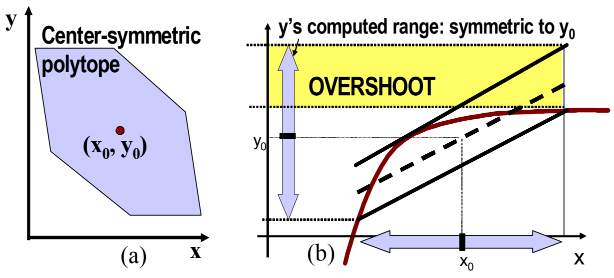

y
Center-symmetric polytope
(x_0, y_0)
(a)
x
y's computed range: symmetric to y_0
OVERSHOOT
y_0
(b)
x_0
x

FIGURE 1. (a) Joint range U^{xy} of a pair of affine intervals is a center-symmetric polytope. (b) Symmetric affine form symmetrically bounds range of any nonlinear affine operator, such as this *log*( ) calculation.

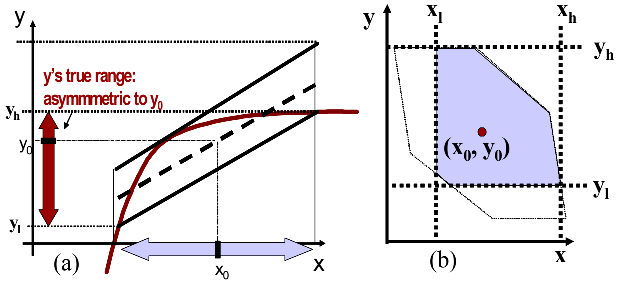


FIGURE 2. (a) Eliminating overshoot in *log*() using asymmetric enforced bounds. (b) Impact on joint-range polytope of affine forms.

erands along with a new error symbol to account for the error. For the important cases of multiplication and division, the affine approximations suggested by [6] are too pessimistic. To reduce this impractical pessimism, and support better bounds for arbitrary binary operations, we introduce the *minvolume* approximation.

We approximate any non-affine binary operation $z = f(x, y)$ as

$$\hat{z} = A\hat{x} + B\hat{y} + C + D\varepsilon \tag{3}$$

where ε is a new error symbol and A, B, C, D are constants to be determined for the least error. Geometrically, this can be visualized as finding two parallel planes in (x, y, z) space, such that the volume between them encloses the surface of z over the domain given by $U^{xy} = \{[x], [y]\}$, as tightly as possible. This can be formulated as

$$\begin{gathered} \mathit{minimize}\ C_1 \ \ \mathit{and} \ \ \mathit{maximize}\ C_2 \\ z \geq Ax + By + C_2,\ \ z \leq Ax + By + C_1 \\ C_1 > C_2,\ \ x, y \in U^{xy} \\ \hat{z} = A\hat{x} + B\hat{y} + (0.5)(C_1 + C_2) + (0.5)(C_1 - C_2)\varepsilon \end{gathered} \tag{4}$$

We seek to compute two planes such that they are as close as possible while still bounding all *relevant* values of z. To simplify the problem, we fix A, B to be the partial derivatives at the center point

$$A = \frac{\partial}{\partial x} z(x_0, y_0),\ \ B = \frac{\partial}{\partial y} z(x_0, y_0) \tag{5}$$

Then, the solution is

$$\begin{gathered} C_1 = max(d(x, y)) \\ C_2 = min(d(x, y)) \\ d(x, y) = z - (Ax + By) \\ x, y \in U^{xy} \end{gathered} \tag{6}$$

Here $d(x, y)$ is the *distance function*. Once the affine form $\hat{z}$ is determined, the next step is to determine the bounds $[z]$ on the result.

To give a concrete example of the idea, will show how this works for multiplication. Division is handled in a similar fashion. (See [12][15] for details.) For multiplication, the values of A and B are

$$A = \frac{\partial}{\partial x}(x \cdot y)(x_0, y_0) = y_0,\ \ B = \frac{\partial}{\partial y}(x \cdot y)(x_0, y_0) = x_0 \tag{7}$$

The distance function, in terms of the intervals, is given as

$$dist(\hat{x}, \hat{y}) = \hat{x} \cdot \hat{y} - (y_0\hat{x} + x_0\hat{y}) = (\textstyle\sum_i x_i\varepsilon_i)(\sum_i y_i\varepsilon_i) - x_0 y_0 \tag{8}$$

The parameters C_1, C_2 are given by the extreme values of the distance function over the joint range of x and y. Let us now define

$$u = x - x_0 = \textstyle\sum_i x_i\varepsilon_i,\ \ v = y - y_0 = \sum_i y_i\varepsilon_i \tag{9}$$

Then, from (8), the distance function reaches its extremes when uv reaches its extremes, over U^{uv}, the joint range of u, v. Note that U^{uv} has the same shape as U^{xy} (i.e., the possibly truncated polytope of Fig. 2), but is centered at $(0, 0)$. It is easy to show that the extreme values of uv can only occur on the boundary of any closed region U^{uv}. Given this, we need scan only the edges of the polygon, and that too only at three points on any edge: the two end points and at $v = c/(2b)$, where the equation of the edge line is $au + bv = c$.

Having found the extrema of uv as $p = min(uv),\ \ q = max(uv)$. it is straightforward to complete the affine approximation

$$\hat{z} = y_0\hat{x} + x_0\hat{y} - x_0 y_0 + (p + q)/2 + (q - p)\varepsilon/2 \tag{10}$$

The next step is to compute the enforced, possibly asymmetric bounds z_l and z_h, where

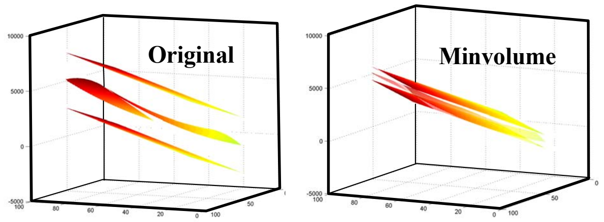


FIGURE 3. Comparison of affine approximations for multiplication of two affine intervals: $\hat{x} = 50 + 25\varepsilon_1 + 25\varepsilon_2$, $\hat{y} = 50 + 25\varepsilon_1 - 25\varepsilon_2$ Bounding planes appear above/below the actual nonlinear product range.

$$z_l = min(xy),\ \ z_h = max(xy) \tag{11}$$

This is easily done by scanning the boundary of U^{xy}, just as we did for uv. Given that each side of the polygon U^{xy} is a straight line

$$y = y_0 + ([c - a(x - x_0)]/b) \tag{12}$$

there are three points on the edge that are of interest: the two end points and one internal point $x = (ax_0 + by_0 + c)/2a$. Fig. 3 shows a comparison between the original affine [6] and our minvolume approximations for one multiplication operation.

5. Tighter Bounds: Probabilistic Bounding

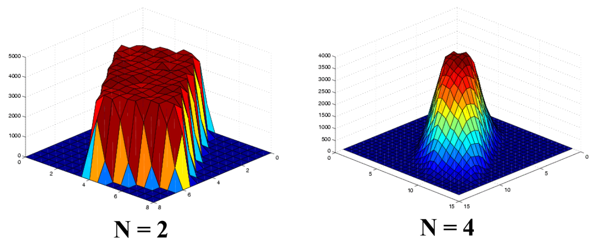
Until now, our extensions to the affine form have been purely deterministic improvements to the tightness of coefficient/bound calculations. To proceed further, we need to make some concrete assumptions about the statistics of the ε_i terms. So, let us take the obvious first step, and regard these as independent, identically distributed (iid) *uniform* random variables.

Consider the case of a single affine form, x. As the number of uncertainty terms, N increases, we can exploit the *Central Limit Theorem* argument of [9]. Under a rather broad set of conditions, the resulting distribution of x should tend towards a normal distribution (even when the ε_i are not iid, under the assumption that no single variation source dominates [13]). Hence, much of the mass of the distribution lies within a range smaller than that indicated by the *radius* of the interval, which is the size of the largest deterministic excursion from the nominal, easily computed as

$$R(\hat{x}) = \textstyle\sum_1^N |x_i| \tag{13}$$

In other words, the chance of x taking a value far from its nominal value is extremely small. We can exploit this to estimate tighter *probabilistic bounds* for the interval. These bounds are just *confidence intervals* from probability theory [12]. For example, if we are satisfied with accounting for 99.7% ($\pm 3\sigma$ range for a normal distribution) of the possible values of x, we can use the 99.7% confidence interval to tighten the bounds of x. These bounds are implemented as the enforced bounds x_l, x_h from Sec. 3. If there exist bounds due to asymmetry (Fig. 2), the tightest combination of bounds is chosen.

The problem, of course, comes when we try to apply these ideas to the more complex case of arbitrary binary interval operations. We now need to look at the joint range U^{xy} of the two arbitrary intervals $\hat{x}$, $\hat{y}$. As the number of uncertainty terms, N, increases, not only do the two marginal distributions tend towards the normal, but also their joint distribution tends toward a joint normal distribution. The probability of a result with values near the edges of the classical affine polytope U^{xy} falls off rapidly (Fig. 4). Hence, the polytope



N = 2 N = 4

FIGURE 4. Joint distribution in a 2-D interval: rapid decay of probability of values toward edges of U^{xy} bounding polygon.

U^{xy} can be approximated as the region bounded by a *confidence ellipse*, given the required confidence level (e.g. 99.7%). A joint normal distribution is given by the following density function:

$$f(x, y) = \frac{e^{\left[-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]\right]}}{2\pi\sigma_x\sigma_y\sqrt{(1-\rho^2)}} \tag{14}$$

μ is the mean, σ the standard deviation, and ρ the correlation coefficient. μ and σ are estimated as

$$\mu_x = x_0, \quad \sigma_x = max(x_h - x_0,\ x_0 - x_l) \tag{15}$$

Given, a confidence level λ (the probability of a sample lying within the confidence ellipse), the ellipse equation is

$$\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \left(\frac{y-\mu_y}{\sigma_y}\right)^2 = -2log(1-\lambda)(1-\rho^2) \tag{16}$$

This confidence ellipse approximation of U^{xy} has two very desirable features, for increasing N. First, it defines a significantly tighter region of interest. Second, it makes the minvolume computation more efficient. The complexity of the necessary polygon construction/traversal is $O(MN)$, where M is the number of errors symbols shared by the two variables [12]. With large values of N and M, the computation become prohibitively expensive. Using a single ellipse equation makes the computation much more tractable, with acceptable accuracy even for values of N as small as 5.

To give one concrete example of the application of this probabilistic bounding idea to the case of binary interval operators, we sketch the development of the minvolume approximation for the confidence ellipse representation of U^{xy}. Division is similar, but more complex, and space does not permit.[1]

Again, looking at the $u-v$ space, U^{uv} is now represented as an ellipse with enforced bounds on it (Fig. 5). Suppose the boundary ellipse equation is

$$au^2 + bv^2 + cuv = 1 \tag{17}$$

$$\therefore\ uv = -\frac{c}{2a}v^2 \pm \frac{v}{2a}\sqrt{(c^2 - 4ab)v^2 + 4a} \tag{18}$$

We can solve for the extreme values of uv (which must again lie on the boundary ellipse) by solving

$$\frac{d}{dv}(uv) = 0 \tag{19}$$

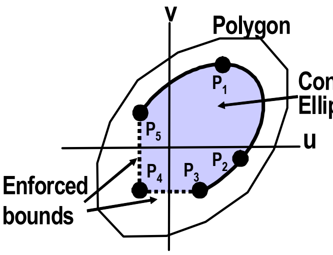


FIGURE 5. Tracing the confidence ellipse boundary for minvolume approximation, with enforced bounds: P_3-P_5 are the endpoints of the bounding edges, P_1, P_2 are the roots within the bounds

This involves solving a quartic equation, leading to four roots. Hence, only these four points on the ellipse need to be checked. If there are any enforced bounds, only those roots of (19), that satisfy these bounds, need to be checked, along with the end points of the bounding edges. In the example shown in Fig. 5, the extrema of uv would be at two out of the five points shown.

The bounds for the result (z_l, z_h) are computed similarly by checking points along the boundary of U^{xy}, which has exactly the same shape as U^{uv}.

The discussion till now focussed on uniformly distributed random variations, i.e., uniform ε_i. However, in real world scenarios (especially in the case of IC manufacturing), we often see variations that have normal distributions. We denote a normal distribution as $N(\mu, \sigma)$, where μ is the mean, and σ the standard deviation.

Our methods can also be extended to the case where the error symbols represent normally distributed variations. The representation of the variable is the same as in (2), but let us assume that each weighted error symbol now represents a normally distributed random variable with standard deviation x_i ($N(0, x_i)$). How can our affine form $(\hat{x}, [x])$ be used to represent some finite range approximation of the probability distribution of a variable x, which no longer has finite support?

Ideally, we want to use our enforced bounds ($[x]$) to mark the confidence interval for x (assume the confidence level used is 99.7%). In the case that the enforced bounds are symmetric, we have a standard normal distribution

$$x \in N\left(x_0, \sqrt{\textstyle\sum_{i=1}^{N} x_i^2}\right) \tag{20}$$

and we can easily assign the enforced bounds to the proper values for a standard 3σ confidence level:

$$\begin{gathered} var(\hat{x}) = 3\sqrt{\textstyle\sum_{i=1}^{N} x_i^2} \\ x_l = x_0 - var(\hat{x}), \quad x_h = x_0 + var(\hat{x}) \end{gathered} \tag{21}$$

The resulting distribution is normal

$$x \in N\left(x_0, \sqrt{\textstyle\sum_{i=1}^{N} x_i^2}\right) \tag{22}$$

However, after any non-affine operation (like multiply), our enforced bounding information $[x]$ can no longer be computed by (21), and may not even be symmetrical. The distribution is no longer normal; each x_i can be interpreted as approximating the relative sensitivities of x to the variation sources and also the correlation between different intervals, (e.g., just as in prior statistical static

[1] In particular, the defining ellipse equation is now of 6th instead of 4th order, and necessitates a heuristic solution. However, a classical result on the distribution of the quotient of normals [14] can be used to improve the computation of the required enforced bounds. See [12], [15].

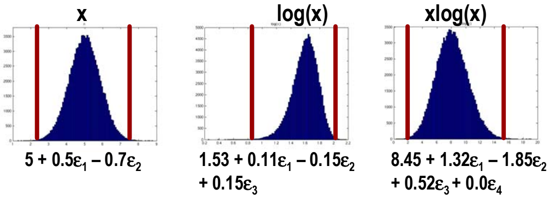


FIGURE 6. Illustration of enforced bounds (vertical lines) and non-affine operations (log, multiply), with normally distributed error symbols.

timing formulations [1]). Also, the enforced bounds are no longer exact, but approximations of the desired confidence intervals.

We need a way to reinterpret the minvolume approximation now that we cannot easily compute an exact confidence interval. We can borrow ideas from statistical timing again, and try a *variance matching* heuristic. We can alter the computation of the coefficient D of the new error symbol ε in (3) as follows

$$D = \begin{cases} 0, & if \quad var(A\hat{x} + B\hat{y}) \geq max(z_h - z_0, z_0 - z_l) \\ \dfrac{\sqrt{max(z_h - z_0, z_0 - z_l)^2 - var(A\hat{x} + B\hat{y})^2}}{3}, & else \end{cases} \tag{23}$$

Thus, D is computed so as to match a sensible model of the "radius" (21) of the resulting affine form ($\{\hat{z}, [z]\}$) to the maximum extent of z on either size of its central value.

Before we proceed to a more extensive discussion of results, let us illustrate in Fig. 6 how all these ideas come together with one quick example. Consider a variable x, and the computation $xlog(x)$. Suppose x has two sources of normally distributed variation ($\varepsilon_1, \varepsilon_2$). The enforced bounds (shown as vertical lines in the figure) are symmetric and, as in (21), coincide with the $\pm 3\sigma$ points of the distribution of x, which is itself normal. After the $log(x)$, the enforced bounds are no longer symmetric and we use (23) instead of (21). The distribution is no longer normal. Note that a new error symbol has been added to account for the variation not accounted for by the original error symbols. The next operation is a multiply to compute $xlog(x)$. Once again, the resulting distribution is not normal and the enforced bounds are asymmetric.

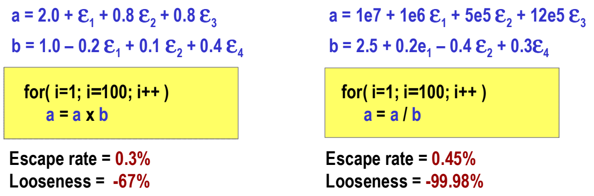
The error symbol coefficients provide an approximate measure of the correlation between two affine variables, as shown in Table 1. Our interval model, while not tracking any explicit model of the pdf, is doing a good job of tracking the range and essential correlations.

6. Experimental Results

We present here three categories of results: (i) simple operator-level experiments to show the power of the probabilistic model for very nonlinear operations; (ii) sparse matrix operations, inspired by the strategies used for modern power grid analysis [16]; and graph-based variational delay analysis, inspired by statistical timing analysis [1]. All these applications are relatively abstract; our goal in this paper is to focus first on the success of the numerics, at a fine-grain level. (Additional details appear in [12][15].)

Correlation Pair	Intervals	Monte Carlo
x , $log(x)$	0.877	0.991
x , $xlog(x)$	0.975	0.999
$log(x)$, $xlog(x)$	0.962	0.984

TABLE 1. Correlation between variables in the computation chain of Fig. 6, comparing probabilistic intervals and Monte Carlo simulation



$a = 2.0 + \varepsilon_1 + 0.8\ \varepsilon_2 + 0.8\ \varepsilon_3$

$b = 1.0 – 0.2\ \varepsilon_1 + 0.1\ \varepsilon_2 + 0.4\ \varepsilon_4$

```
for( i=1; i=100; i++ )
    a = a x b
```

Escape rate = 0.3%
Looseness = -67%

$a = 1e7 + 1e6\ \varepsilon_1 + 5e5\ \varepsilon_2 + 12e5\ \varepsilon_3$

$b = 2.5 + 0.2e_1 – 0.4\ \varepsilon_2 + 0.3\varepsilon_4$

```
for( i=1; i=100; i++ )
    a = a / b
```

Escape rate = 0.45%
Looseness = -99.98%

FIGURE 7. Performance of deep multiple/divide chains, uniform error symbols, measured across 10 runs of 100,000 Monte Carlo samples

To assess both accuracy and speed, we compare against standard Monte Carlo analysis, assuming either uniform or normally distributed error symbols, as from Sec. 5. Suppose we represent the Monte Carlo simulated range for a result as $z \in [z_l^s, z_h^s]$. We define two metrics to quantify the accuracy of the result computed using probabilistic intervals -- the *Looseness* (*l*) and the *Escape Rate* (*e*):

$$l = 100 \times \left(\frac{z_h - z_l}{z_h^s - z_l^s} - 1 \right) \qquad e = 100 \times \left(1 - \frac{N_{int}}{N_{sim}} \right) \tag{24}$$

N_{sim} is the number of Monte Carlo samples and N_{int} is the number of samples that lie within the computed bounds $[z_l, z_h]$ from our probabilistic interval-valued result. Both quantities are expressed as percentages. The *escape rate* tells us what fraction of our Monte Carlo samples failed to lie within the bounds predicted by our interval; we prefer this answer to be ~0%. The *looseness* tell us the ratio of the sizes of the Monte Carlo interval and the computed probabilistic interval; 0% means they are identical, and the sign tells us if our computed interval is smaller (neg) or larger (pos).

Let us begin with an operator-level "stress test", focusing on our improved multiplication/division operators. Fig. 7 shows a pair of deep, iterated multiply/divide chains. These are stressful since the ranges of the results grow/shrink dramatically, as does the dynamic range of the individual error symbol coefficients. Nevertheless, we see escape rates well under 1%. It is interesting to note the negative looseness metric here: -99.98% for the divide result means we are capturing 99.55% of the Monte Carlo samples, in a probabilistic interval that is just 0.02% as wide as the Monte Carlo sample range.

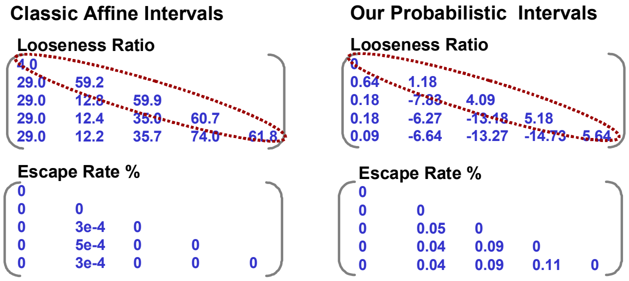
To compare the classical affine model and our improved model directly, let us examine a small matrix problem: a dense Cholesky decomposition. A 5x5 matrix is small enough to inspect visually, but large enough to challenge. We construct the matrix as

$$A = [a_{ij}], \quad i, j \in \{1, 2, 3, 4, 5\}$$
$$a_{ij} = \begin{cases} 5i^2, & i = j \\ min(i, j), & i \neq j \end{cases} \tag{25}$$

Each element has its own error symbol with 30% variation (uniform distribution). We average over 10 runs, 100K samples each.

Results appear in Fig. 8. We see improvements in looseness of up to 10X. The classical affine computation is exceptionally conservative, as can be seen from the escape number. The price of our tighter intervals is some nonzero escape, but we are still well under 1% for this toy example.

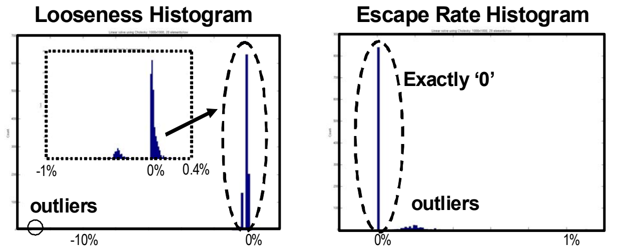
Fig. 9 shows results from a more realistic example: Cholesky decomposition of large, symmetric, sparse, 1000x1000 matrix, 1 run of 10K samples. The off-diagonal non-zero elements are chosen


Classic Affine Intervals
Our Probabilistic Intervals
Looseness Ratio
Looseness Ratio
Escape Rate %
Escape Rate %

$$\begin{pmatrix} 4.0 & & & & \\ 29.0 & 59.2 & & & \\ 29.0 & 12.8 & 59.9 & & \\ 29.0 & 12.4 & 35.0 & 60.7 & \\ 29.0 & 12.2 & 35.7 & 74.0 & 61.8 \end{pmatrix} \quad \begin{pmatrix} 0 & & & & \\ 0.64 & 1.18 & & & \\ 0.18 & -7.83 & 4.09 & & \\ 0.18 & -6.27 & -13.18 & 5.18 & \\ 0.09 & -6.64 & -13.27 & -14.73 & 5.64 \end{pmatrix}$$

$$\begin{pmatrix} 0 & & & & \\ 0 & 0 & & & \\ 0 & 3e-4 & 0 & & \\ 0 & 5e-4 & 0 & 0 & \\ 0 & 3e-4 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & & & & \\ 0 & 0 & & & \\ 0 & 0.05 & 0 & & \\ 0 & 0.04 & 0.09 & 0 & \\ 0 & 0.04 & 0.09 & 0.11 & 0 \end{pmatrix}$$

FIGURE 8. Perfomance of 5x5 dense Cholesky decomposition, with uniformly distributed variation sources, averaged over 10 runs of 10,000 Monte Carlo samples



Element-wise Looseness (%)
(Min: -13.3%, Max: 0.32%)

Element-wise Escape Rate (%)
(Min: 0% Max: 1.1%)

FIGURE 9. Performance of 1000x1000 sparse Cholesky decomposition, measured over ~1043 nonzeroes of final matrix. 1 run, 10,000 Monte Carlo samples. CPU times: 6.6s for intervals, vs. 1152s

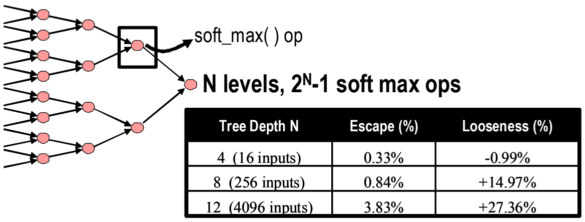
randomly, such that there are approximately 20 elements per row. The non-zero values are assigned as in (25). Each element has its own error symbol with 30% variation (uniform distribution). Results are again quite good. Most of the element-wise escape numbers are 0%. Our probabilistic intervals are mainly at 0% looseness, element-wise, with a few slightly smaller or large than the actual sampled range. Doing this computation with the classical affine intervals produces uselessly loose intervals. The CPU time savings is also attractive: 6.6s versus 1152s on a 2GHz Pentium.

Let us next consider a graph algorithm in the style of statistical static timing calculations. We wish to calculate the range of delay uncertainty at the output of a binary tree of randomly varying delay elements. To make this more challenging, we use the *soft-max* operator, a common smooth approximation of the max function

$$softmax(x, y) = \frac{1}{k}log(e^{kx} + e^{ky}) \tag{26}$$

k is a shaping factor: the larger it is, the better is the approximation. The value chosen is 4. We construct a converging binary tree of 2-input soft-max operators. The inputs at the leaves are affine interval forms. For a depth of N, the total number of unique error symbols at the inputs is N^2. These error symbols are randomly assigned to the inputs, such that each input has N symbols on average. The central values of the inputs are assigned randomly from $[20, 30]$ and the error symbol coefficients are assigned randomly from $[-5, 5]$. The error symbols now represent *normally* distributed random variables. The results are averaged over 10 runs of 10K samples each.

The alternating layers of *exp*( ) "amplification" and *log*( ) "attenuation" make this a difficult test for any interval-valued method. Results appear in Fig. 10. Given the difficulty, we see a larger, but still acceptable escape rate. Looseness degrades some, however, because of the difficulty of capturing perfect confidence intervals over these very non-linear computations.



Tree Depth N	Escape (%)	Looseness (%)
4 (16 inputs)	0.33%	-0.99%
8 (256 inputs)	0.84%	+14.97%
12 (4096 inputs)	3.83%	+27.36%

FIGURE 10. Performance of graph delay computation in soft-max trees of varying size. 10 runs of 10,000 Monte Carlo samples.

7. Conclusions

We have extended the classical theory of affine intervals to support asymmetric interval ranges, tighter range bounds via the unifying concept of the minvolume approximation, and a more rigorous confidence-interval style statistical interpretation. Preliminary results are encouraging: we can achieve intervals with bounds up to 10X tighter than standard affine results. Comparison against Monte Carlo simulation are both tight, and typically 10-100X faster. Remaining challenges are algorithms with extremely long computational depth, e.g., iterative matrix solvers, in contrast the direct methods we showed here. These can have chains of computations that "touch" the same variable millions of times. Such algorithms still defy interval analysis, but may yet yield to improved interval methods.

Acknowledgements

We thank Sani Nassif of IBM for valuable discussions in the formative stages of this research. This work was funded by the MARCO Focus Center for Circuit & System Solutions (C2S2) and the Semiconductor Research Corp.

References

[1] C. Visweswariah, et al, "First-order Incremental Block-based Statistical Timing Analysis," *Proc. ACM/IEEE DAC*, June 2004.
[2] M. Orshansky et al, "Fast Statistical Timing Analysis Handling Arbitrary Delay Correlations," *Proc. ACM/IEEE DAC*, June 2004.
[3] M. Mani et al, "An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints," *Proc. ACM/IEEE DAC*, June 2005.
[4] X. Li et al, "Asymptotic Probability Extraction for Non-normal Distributions of Circuit Performance," Proc. ACM/IEEE ICCAD, Nov. 2004.
[5] R.E. Moore, *Interval Analysis,* Prentice-Hall, 1966.
[6] L. H. de Figueiredo and J. Stolfi, "Self-validated numerical methods and applications", Brazilian Mathematics Colloqium monograph, IMPA, Rio de Janeiro, July 1997.
[7] C.L. Harkness, et al, "Interval Methods for Modeling Uncertainty in RC Timing Analysis," *Proc. ACM/IEEE ICCAD*, Nov. 1992.
[8] A. Lempke, et al, "Analog Circuit Sizing based on Formal Methods using Affine Arithmetic," Proc. ACM/IEEE ICCAD, Nov. 2002.
[9] F. Fang et al, "Toward Efficient Static Analysis of Finite-Precision Effects in DSP Applications via Affine Arithmetic Modeling.", Proc. ACM/IEEE DAC, June 2003.
[10] J.D. Ma, et al, "Interval-Valued Reduced Order Statistical Interconnect Modeling," Proc. ACM/IEEE ICCAD, Nov. 2004.
[11] J.D. Ma, et al, "Fast Interval-Valued Statistical Interconnect Modeling and Reduction," Proc. ACM ISPD, Apr. 2005.
[12] C. F. Fang, *Probabilistic Interval-Valued Comuptation: Representing and Reasoning about Uncertainty in DSP and VLSI Design*, PhD. Diss., Carnegie Mellon Univ., 2005.
[13] W. Feller, "An Introduction to Probability Theory and Its Applications", Vol. 2, 3rd ed., Wiley, New York, 1971
[14] G. Marsaglia, "Ratio of Normal Variables and Ratios of Sums of Uniform Variables", *J. of Amer. Stat. Assn.*, 60(309), Mar 1965
[15] A. Singhee, et al, "Probabilistic Interval-Valued Computation: Toward a Practical Surrogate for Statistics Inside CAD Tools," CMU CSSI Technical Report manuscript, in progress.
[16] G. Steele et al, "Full Chip Verification Methods for DSM Power Distribution Systems," *Proc. ACM/IEEE DAC*, 1998.