Interval Computations
No 2, 1993

Automatically Verified Reasoning with Both Intervals and Probability Density Functions

Daniel Berleant

Information about a value is frequently best expressed with an *interval.* Frequently also, information is best expressed with a *probability density function.* We extend automatically verified numerical inference to include combining operands when both are intervals, both are probability density functions, or one is an interval and the other a probability density function. This technique, termed the *automatically verified histogram method*, uses interval techniques and forms a sharp contrast with traditional Monte Carlo methods, in which operands are all intervals or all density functions, and which are not automatically verifying.

Автоматически проверяемые рассуждения с использованием интервалов и функций плотности вероятности

Д. Берлеант

Информация о значении величины часто лучше всего может быть выражена с помощью *интервала*, а также и с помощью *функции плотности вероятности.* Мы обобщаем автоматически проверяемый численный вывод таким образом, чтобы включить случай комбинированных операндов, то есть случай, когда оба операнда являются интервалами, или оба функциями плотности вероятности, или когда один является интервалом, а другой — функцией плотности вероятности. Этот метод, называемый *методом гистограмм с автоматической проверкой*, использует интервальную технику и резко отличается от традиционного метода Монте-Карло, в котором все операнды являются либо интервалами, либо функциями плотности вероятности, и в котором отсутствует автоматическая верификация.

1 Introduction

Accurate and precise numerical information is often unavailable. Therefore we wish to be able to reason with the less exact information that is available. Frequently such information about a value is in the form of an *interval* bounding an actual but unknown value. Frequently also, that information is in the form of a *probability density function,* which describes the relative likelihoods of what the value might be.

One important property of interval mathematics is its ability to support automatically verified — hence correct — inference in the presence of uncertain values. We extend automatically verified numerical inference to include cases where input values may be intervals, or probability density functions, or some inputs may be intervals and others probability density functions.

The method described, called the *automatically verified histogram method,* uses interval techniques. The automatically verified histogram method is compared to traditional Monte Carlo methods, which disallow combining interval operands with density function operands, and which do not provide automatic verification.

2 Operations on intervals: a probabilistic view

In many real world problems, numerical values are not precisely known. In many such cases, an *interval* may be used to bound the range of belief about what the constant value could be [1]. Probabilistically, such an interval constitutes a statement that we are modeling the constant (but unknown) value as being within the bounds of the specified interval with probability 1. In other words, the interval has a probability mass of 1. If we wish to apply some binary *op*eration $\boxed{op}$ to values $x \in X$ and $y \in Y$, X and Y intervals, to get a result $z = x \boxed{op} y$, we can say that $z \in Z = X \boxed{OP} Y$.[1] Probability $p(z \in Z)$ conforms to $p(z \in Z) = p(x \in X) \times p(y \in Y) = 1 \times 1 = 1$, in

[1] $\boxed{OP}$ is the interval extension of $\boxed{op}$, which might be $+, -, \times, \div$, or any binary operation with a corresponding interval analog defined for X and Y. Unary and other n-ary operations are treated similarly.

this case. In general: $p(x \in X) \in [0, 1]$, $p(y \in Y) \in [0, 1]$, and

$$p(z \in Z) = p(x \in X)p(y \in Y). \quad (1)$$

Equation 1 requires two assumptions:

1. x and y are independent. If there is some degree of dependency, much less can be said about $p(z \in Z)$ unless the dependency is characterized, and even then equation (1) will not hold in most cases.

2. Operation $\boxed{OP}$ avoids introducing excess width. If excess width could occur, then the equation (1) is weakened to $p(z \in Z) \geq p(x \in X)p(y \in Y)$.

3 Operations on probability density functions using interval arithmetic

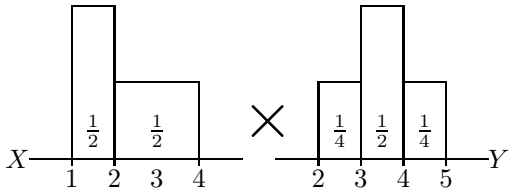
The probabilistic view of interval operations above leads naturally to an existing histogram discretization algorithm called the *histogram method* for doing operations on probability density functions (PDFs). The histogram method discretizes PDF operands using intervals, uses interval operations to generate intermediate results, and then constructs a result PDF. This technique, also known as "discrete combination of random variables" was first described by Ingram et al. in 1968 [2] and further developed by Colombo and Jaarsma in 1980 [3]. It has subsequently generated attention mostly in reliability analysis [4, 5, 6, 7] although the technique itself is a general one. Kaplan's method [8] is a popular variation, generating over 50 citations in Science Citation Index over the years, but it is unclear how to make Kaplan's method automatically verifying. Moore [9] independently developed another variation in which results are expressed as cumulative distribution functions (CDFs). We describe the histogram method next, extending it later into the automatically verified histogram method.

3.1 The histogram discretization algorithm

In the histogram method, PDFs are discretized using histograms. Each histogram bar is characterized both by an interval describing its placement

on the real number line and by a probability mass. To operate on a pair of PDFs X and Y, their histogram discretizations are combined as follows.

1. Compute the Cartesian product of the bars of the histograms describing X and Y.
2. For each member (X_i, Y_j) in the Cartesian product, produce an *intermediate result interval* by:
   (a) executing the corresponding interval arithmetic operation on X_i and Y_j to get $Z_{ij} = X_i \boxed{OP} Y_j$; then
   (b) associating with Z_{ij} the probability $p(Z_{ij}) = p(X_i)p(Y_j)$, in accordance with equation 1.
3. The intermediate result intervals are each part of an *intermediate result collection,* exemplified by the table in Figure 1. The intermediate result intervals may be combined to get a final result, as follows.
   (a) Decide on a set of intervals partitioning the domain of Z. This partition determines the placement of the bars in a histogram approximating the distribution function. The particular partition is unspecified by the algorithm, but few bars will tend to provide coarse results.
   (b) Calculate the area (i.e. the probability mass) for each histogram bar of Z defined by the partition (Figure 1), as follows.
      i. Any intermediate result interval Z_{ij} that falls completely within some member of the partition has its entire probability mass assigned to the bar corresponding to that member.
      ii. Any intermediate result interval that overlaps more than one member of the partition has its probability mass divided among them, with mass assigned to each partition member in proportion to the fraction of the intermediate result interval it overlaps. For example, intermediate result interval #1 in the table of Figure 1 is $[2, 6]$ with probability $\frac{1}{8}$. The partition of the domain of Z (Figure 1, bottom) contains intervals $[2, 5]$ and $(5, 8]$, so $[2, 5]$ is assigned $\frac{3}{4}$ of the $\frac{1}{8}$ probability, or $\frac{3}{32}$, because $\frac{3}{4}$ of the width of intermediate result interval $[2, 6]$ overlaps $[2, 5]$. Similarly $\frac{1}{4}$ of the width of



Cartesian product term #	Bar intervals and masses	Operand 1 (X_i)	Operand 2 (Y_j)	Intermediate result collection intervals (Z_{ij})
#1	interval probability	$[1, 2]$ 1/2	$[2, 3]$ 1/4	$[2, 6]$ 1/8
#2	interval probability	$[1, 2]$ 1/2	$[3, 4]$ 1/2	$[3, 8]$ 1/4
#3	interval probability	$[1, 2]$ 1/2	$[4, 5]$ 1/4	$[4, 10]$ 1/8
#4	interval probability	$[2, 4]$ 1/2	$[2, 3]$ 1/4	$[4, 12]$ 1/8
#5	interval probability	$[2, 4]$ 1/2	$[3, 4]$ 1/2	$[6, 16]$ 1/4
#6	interval probability	$[2, 4]$ 1/2	$[4, 5]$ 1/4	$[8, 20]$ 1/8

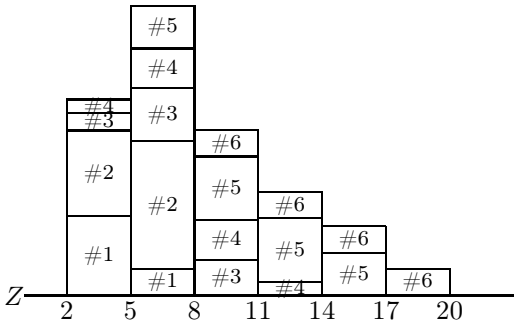


Figure 1: **Multiplication of two histograms.** The Cartesian product of the 2 bars in X and the 3 bars in Y leads to an intermediate result collection containing 6 intermediate result intervals. A result histogram for $Z = X \times Y$ was defined to have several bars, each with a width of 3 and placed from 2 to 20 on the horizontal axis. The mass of each intermediate result interval was divided among the bars of the result histogram, based on the mass of the intermediate result interval and what proportion of its width overlaps with a given histogram bar in the PDF for Z. The contribution of each intermediate result interval to a bar of the result histogram is indicated by printing its number in a section of the bar, with the size of the section indicating the probability mass contributed.

$[2, 6]$ overlaps $(5, 8]$, which is therefore assigned probability mass $\frac{1}{4} * \frac{1}{8} = \frac{1}{32}$. This process will be termed *proportional assignment.*[2]

iii. All of the probabilities assigned to a partition member are summed to give the total probability of the member. This is done for each partition member.

(c) The probability of each bar equals its area, so its height h can be calculated from $h = \dfrac{\text{area}}{\text{width}} = \dfrac{\text{probability}}{\text{width}}$ and the histogram can be drawn, as at the bottom of Figure 1.

While the example in Figure 1 happens to be of multiplication, many different operations and functions of two variables can be used to get an intermediate result collection. If the calculation produces an intermediate result collection whose intervals may have excess width, then proportional assignment (item 3(b)ii above) will tend to cause the result PDF to spread out. Dependencies between operands may cause even less predictable distortion in the result PDF.

So far, the algorithm is essentially as described by Ingram et al. [2]. Colombo and Jaarsma's further development [3] uses histogram bars of varying width but constant mass, as does A. S. Moore [10]. Kaplan's variation [8] approximates the bars with their *midpoints* and probability masses. It is unclear how Kaplan's variation could be made automatically verifying. R. E. Moore [9] and A. S. Moore [10] foreshadow the present paper by expressing results as CDFs. R. E. Moore discretizes results more wisely than the previous (and the present) work. Both R. E. Moore and A. S. Moore apply their methods to non-trivial problems. Unlike the present paper they do not address automatic verification.

While the histogram method has an established place in the literature, it produces approximations and so is not automatically verifying. The approximating character of the algorithm is due to two problems:

1) discretizing a PDF into a histogram seems at first glance to produce merely an approximation of the PDF; and

[2]Since we are working with probability masses, whether the interval is open or closed is irrelevant for PDFs not containing impulses.

2) the proportional assignment step makes assumptions about how the probability mass of an intermediate result interval is distributed over that interval.

These two problems are discussed in turn.

3.2 Automatically verified discretization

Discretizing a PDF as a histogram may at first appear to force a possibly smooth and continuous PDF into the outline or silhouette of a histogram — a piecewise continuous curve with horizontal line segments (the tops of the bars) connected by vertical line segments. This interpretation is both unnecessary and, from the perspective of correctness, highly undesirable. An appropriate change in our interpretation of what a histogram means allows us to view the histogram as *correct,* rather than a (very likely) incorrect approximation. Let us elaborate.

3.2.1 The histogram representation as correct, not approximate

Observe that here a histogram defines:

- a set of non-overlapping intervals, and for each member interval I_j,
- the probability $p(I_j)$ that the uncertain variable's value is in I_j.

Note that no assumption is required about *how* the probability mass $p(I_j)$ is distributed over I_j. Any apparent flatnesses in the outline of the histogram are due to an artifact: the graphical representation used to show histograms, which depicts bars with flat tops. In fact, a histogram representation of a PDF actually corresponds to *any* PDF which has the same probability masses over the intervals specified by the histogram bars as do the bars themselves. To maintain correctness, we need simply adopt the reasonable and useful interpretation that *the apparent flat tops of the bars are for graphical purposes only and a histogram bar leaves undefined how its probability mass is actually distributed over its interval.* Figure 2 illustrates this by showing some obviously different PDFs that are correctly represented by the same histogram.

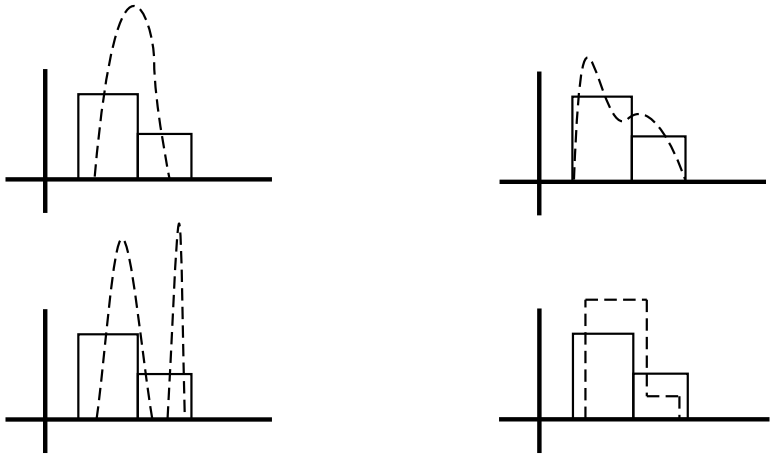


Figure 2: A few of the infinite number of probability density functions corresponding to the same two-bar histogram. The histogram partitions its domain into two intervals. Each histogram bar represents an interval I_j, and also has a height which encodes an area and hence a probability mass. This area equals the probability mass over I_j of each PDF above.

A histogram which is interpreted this way is not really a PDF, because a PDF fully defines how probability mass is distributed over its domain. In contrast, the histogram only partially defines how probability mass is distributed over its domain. Discretizing a PDF as a histogram, then, involves not an approximation but rather a relaxation in representation, and correctness is maintained (although some information is lost).

Histograms, PDFs, and CDFs. We have just seen how a histogram is a weaker description than a PDF, and correctly describes any PDF in a family of PDFs. This family of PDFs can also be felicitously described as two cumulative distribution functions (CDFs) that *bound* the family of CDFs corresponding to the family of PDFs. The faster rising of the two bounding CDF is obtained by taking the mass of each histogram bar to be concentrated at the low bound of its interval. The slower rising CDF is obtained by concentrating the mass of each bar at its interval's high bound. For each point u in domain x, the pair of CDFs provides bounds for an interval $P(x \leq u) = \left[\underline{p(x \leq u)}, \overline{p(x \leq u)}\right]$ (see Figure 3).

Cartesian product term #	Bar intervals and masses	Operand 1 (X_i)	Operand 2 (Y_j)	Intermediate result collection intervals (Z_{ij})
#1	interval probability	$[1, 2]$ 1/2	$[2, 3]$ 1/4	$[2, 6]$ 1/8
#2	interval probability	$[1, 2]$ 1/2	$[3, 4]$ 1/2	$[3, 8]$ 1/4
#3	interval probability	$[1, 2]$ 1/2	$[4, 5]$ 1/4	$[4, 10]$ 1/8
#4	interval probability	$[2, 4]$ 1/2	$[2, 3]$ 1/4	$[4, 12]$ 1/8
#5	interval probability	$[2, 4]$ 1/2	$[3, 4]$ 1/2	$[6, 16]$ 1/4
#6	interval probability	$[2, 4]$ 1/2	$[4, 5]$ 1/4	$[8, 20]$ 1/8

Integrating the table above (↑) produces the curves below (↓).

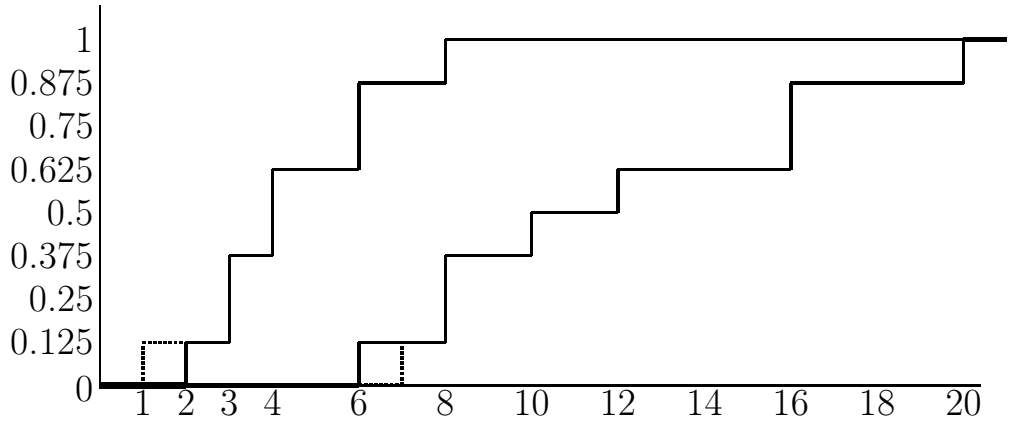


Figure 3: Bounding the family of CDFs coded by an intermediate result collection. If the probability mass of each intermediate result interval is assumed to be concentrated at its low bound, we get the higher CDF curve. If masses are concentrated at the high bounds of the intervals, we get the lower CDF curve. Any other distribution of mass produces a CDF falling somewhere between the two CDFs shown. If interval #1 is widened to $[1, 7]$ (simulating excess width) then the curves shown are widened out to the dotted portions — less constraining but still automatically verified.

3.3 Creating result histograms assumes flat distributions

Let us move from the problem of verified *operands* to the problem of verified *results.*

To create a result histogram, previous work has assumed that the mass of an intermediate result interval can be proportionally assigned to the bars of the result histogram with which it intersects, as in Section 3.1, item 3(b)ii. Unfortunately this proportional assignment assumption is in general unjustified, and usually false. One way to circumvent this problem is to discretize input PDFs using histograms with a large number of narrow bars: as the number of bars in the operands increases toward ∞, the percentage of intermediate result intervals (and the total of their probability masses) involved in proportional assignment calculations decreases toward zero. Unfortunately, at the same time computational cost rises toward ∞. A computationally cheaper way to avoid proportional assignment is simply to refuse to assume how the probability mass is distributed over an intermediate result interval. We will avoid proportional assignment and therefore retain the automatic verification property associated with the intermediate result collection itself. This is elaborated next.

4 Automatically verified operations on PDFs

Observe that an intermediate result collection contains intervals and their associated probability masses, but does not define how the masses are distributed over the intervals. The intermediate result collection is already automatically verified. We need simply process this intermediate result collection in some way which avoids assumptions about the distributions of probability masses within intermediate result intervals, thereby preserving automatic verification. Creating a result histogram requires the assumptions we must avoid, so we cannot create a result histogram. If instead of insisting on ending up with histograms representing result PDFs we are willing to be satisfied with *cumulative distribution functions* (CDFs), which are the integrals of PDFs, we can avoid the unwanted assumptions and thereby retain the automatic verification property of intermediate result collections. An in-

termediate result collection may be represented as a pair of CDFs bounding the family of possible CDFs in the same way a histogram can be represented using bounding CDFs (Section 3.2.1).

The present method is best introduced with an example (Figure 3). Since no assumption may be made about the distribution of mass within any intermediate result interval, of course the integral of the intermediate result collection cannot be fully defined. Instead we bound it with upper and lower CDFs. These CDFs bound a space of CDFs *containing all CDFs corresponding to some distribution of the intermediate result interval masses over their respective intermediate result intervals.*

We now explain the process illustrated in Figure 3 in detail. The intermediate result collection forms a kind of Cartesian product derived from the bars of the operand histograms. The lowest low bound in the intermediate result collection is for Cartesian product term #1 which is an intermediate result interval specifying a probability mass of $\frac{1}{8}$ distributed over the interval $[2, 6]$. If that mass was concentrated at the interval's low bound of 2, then the integral of the intermediate result collection would jump to $\frac{1}{8}$ as soon as the domain value increased past 2. This is a faster rise than any other distribution of $\frac{1}{8}$ mass over the interval $[2, 6]$. Similarly, if the $\frac{1}{4}$ mass of Cartesian term #2 was concentrated at its low bound of 3, the CDF would jump by an additional $\frac{1}{4}$ as the domain value passed 3, and its value at $3+\epsilon$ would now be the sum of the total mass that has been expended so far, $\frac{1}{8} + \frac{1}{4} = \frac{3}{8}$. Continuing this process, we take the masses of the remaining Cartesian product terms to be concentrated at their low bounds as well. Then, the integral of the intermediate result collection rises faster than it would for any other distribution of the masses within the intermediate result collection intervals. The result is the higher of the two CDFs pictured in Figure 3.

To get the *lower* of the CDFs in Figure 3, we take the mass of each Cartesian product term to be concentrated at its *high* bound, instead of its low bound as before. Then the CDF representing the integral of the intermediate result collection rises more slowly than it would for any other distribution of the masses over the intervals in the Cartesian product forming the intermediate result collection. The result is an automatically verified answer: two CDFs bounding the family of CDFs that might be produced from the operand PDFs by the operation.

4.1 Dependencies among input variables

The automatically verified histogram method requires independent inputs, because the probability mass calculations that are an essential part of producing an intermediate result collection use equation 1 which assumes operands are independent: if bar X_i of histogram X has probability $p(X_i)$ and bar Y_j of histogram Y has probability $p(Y_j)$, then $p(X_i)p(Y_j) = p(x \in X \wedge y \in Y)$ only if x and y are independent. Appropriate modification would be needed to extend the automatically verified histogram method to dependent or partially dependent inputs. One appropriate modification would be to calculate each intermediate result interval's probability mass directly from a 2-D histogram of 4-sided bars (X_i, Y_j) describing the distribution of mass over the plane of possibly dependent input variables x and y. Then, the weight of an intermediate result interval would be taken directly from the weight of the 4-sided bar determined by the two operand intervals, rather than by multiplying the weights of the operand intervals as before. Another appropriate modification would be to reformulate the problem in terms of inputs that are independent, as in Moore [9].

4.2 Excess width

As with many applications of interval calculations, excess width may appear due to variables appearing repeatedly in expressions.

With the automatically verified histogram discretization method, the effect of excess width is to enlarge intervals in the intermediate result collection. This in turn enlarges the family of CDFs by causing the higher of the bounding CDFs to rise too quickly, or the lower to rise too slowly, or both. For example, suppose the intermediate result interval of Cartesian term #1 of Figure 3 had excess width sufficient to enlarge both its bounds by 1. Then the interval would be $[1, 7]$ instead of $[2, 6]$ and its mass of $\frac{1}{8}$ could be concentrated as low as 1, leading to a CDF which rises faster over a portion of the domain than for the original low bound of 2. This situation is shown with a dotted line in Figure 3. Similarly, a high bound of 7 instead of 6 means that the mass could be concentrated as high as 7, so that the lower of the two bounding CDFs shown could "wait" until 7 before rising by $\frac{1}{8}$. This means the lower of the bounding CDFs would be rising even slower than before over a portion of its domain. The new, less constraining

portion of the lower bounding CDF is also indicated with a dotted line in Figure 3. To summarize: *Excess width in intermediate result intervals leads to more relaxed bounds on the family of plausible CDFs. This constitutes a weaker but still automatically verified result.*

When expressions consist solely of independent variables occurring once, excess width is not a problem. In other cases, approaches such as various *centered forms* [11, 12] often provide narrower results than naïve evaluation of the interval expression. Supplementing these forms are techniques for removing arbitrary amounts of excess width from expression evaluations. Such techniques are usually described as "computing the range of values" (Moore 1976 [13], Asaithambi et al. 1982 [14], and Cornelius and Lohner 1984 [15]) or as "enclosure methods" (Alefeld 1990 [16]) and have been applied e.g. to electrical circuit tolerance analysis [17]. Artificial intelligence work in this area includes Hyvönen (1992) [18]. Computation time tends to be a problem with these excess width removal techniques.

As is often the case in interval mathematics, excess width can severely weaken the answers obtained. Therefore, it is necessary to assess the quality and usefulness of the results obtained when excess width is present in an application.

5 When some operands are PDFs and others are intervals

So far, we have discussed automatically verified operations when both operands are PDFs. The ideas are easily extended to the case where one operand is a PDF and another is an interval. This is done by using histograms to represent not only PDFs but also intervals. Once both intervals and PDFs are described using histograms, the algorithm developed previously for operating on histograms applies.

We have already seen how a PDF may be represented using a histogram. The alert reader might immediately observe how an interval can be represented as a one-bar histogram:

> Let interval Y be the range of plausible values for y. Then $p(y \in Y) = 1$, although the distribution of that probability

> mass of 1 over Y is undetermined. Recall that here a histogram consists of intervals and the probability mass within each with no assumption about how the probability mass associated with an interval is distributed within the interval. Therefore a single interval with probability 1, such as Y above, may be represented using a one-bar histogram.

We can do automatically verified operations on histograms, and we now know how to describe both PDFs and intervals using histograms. Therefore: *We can do automatically verified operations when both operands are PDFs, both are intervals, or one is an interval and one is a PDF.*

The result of such an operation, as before, is an intermediate result collection. As before, since the distribution of probability masses is not specified completely by an intermediate result collection, integrating it cannot result in a single CDF. Instead, integration produces a family of CDFs, bounded by upper and lower CDFs.

We next apply these ideas to an example.

6 Example: overloading a disk

Consider a simple model of a computer disk filling with data. Data is assumed to flow in with rate $[0.033, 0.047]$ megabytes per hour. Data is deleted, freeing up disk space, at rate $[0.007, 0.012]$ megabytes per hour. The free disk space is initially $[60, 80]$ megabytes. The time t it takes to overload the disk is then described by

$$t = \frac{[60, 80]}{[0.033, 0.047] - [0.007, 0.012]}. \tag{2}$$

We are given some additional information about the free disk space as well: its value is normally distributed within $[60, 80]$ with a mean of 70. The problem is to describe how long it takes for the disk to become completely filled with data. Results were obtained with the help of the Q3 software package [19] and are shown by the thinly drawn outer curves in Figure 4.

The outer curves in Figure 4 are weak. Nevertheless they constitute a stronger result than would be derived by simple conventional means: solving

equation (2) gives $t \in [1500, 3810]$, yet the outer curves provide more information than that, showing for example that overload probably will occur at or after time 1700, and probably will have occurred by time 3400.

Stronger initial data leads to stronger conclusions. When the data inflow rate specification is narrowed from $[0.033, 0.047]$ to $[0.037, 0.043]$ megabytes/hr, the conclusions are correspondingly better (thickly drawn inner curves, Figure 4). In related work, Post and Diltz [20] report on risk analysis of computer systems using pairs of CDFs.

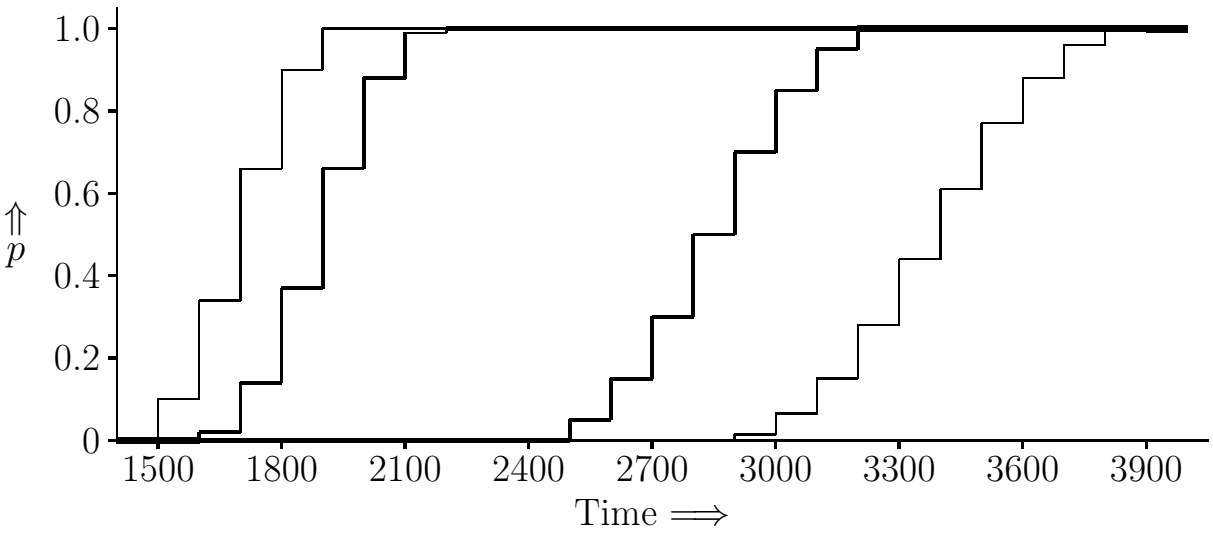


Figure 4: A disk slowly fills with data. The thinly drawn outer CDFs bound the space of possible CDFs that describe the probability of the disk becoming filled with data as time progresses, given a data write rate in $[0.033, 0.047]$. The more thickly drawn inner CDFs bound a smaller family of CDFs that describe the probability of the disk becoming filled over time as well, but with a narrower interval for write rate of $[0.037, 0.043]$. The narrower input led to stronger conclusions, as shown by CDF bounds that are closer together. CDF bounds that are close together constrain the space of possible CDFs describing the probability of disk overload over time more than CDF bounds that are far apart.

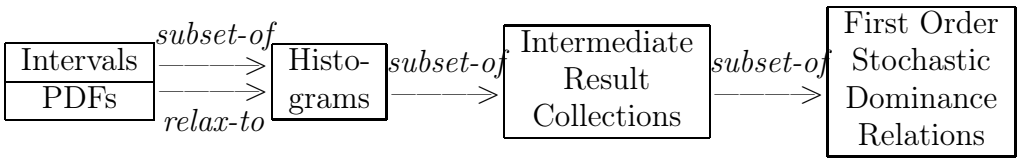


Figure 5: The relationships among intervals, PDFs, histograms, intermediate result collections, and first order stochastic dominance representations. The *subset-of* relations also involve shifts in representational formalism. Going from left to right, none of the relationships above involves approximation, so correctness is preserved.

7 Bounded CDF families and stochastic dominance

The bounding CDFs derived by the automatically verified histogram do not cross, since the lower CDF is as low as possible at every point, and the higher CDF is as high as possible at every point. Further, the two CDFs are different from one another because they are constructed from histograms which in turn are constructed from finite-width intervals each with different possible distributions of mass. Two different, non-crossing CDFs are said to stand in a relationship of *first order stochastic dominance.* Formally,

$$F(x) \leq G(x) \qquad \text{for all } x \tag{3}$$

where F and G are cumulative distributions and the inequality is strict for at least one point in x [21].

The conceptual connections between intervals, PDFs, histograms, intermediate result collections, and first order stochastic dominance relations are shown in Figure 5.

Two CDFs in a first order stochastic dominance relationship bound the family of CDFs consisting of all CDFs that dominate one bounding CDF and are dominated by the other. There are other ways to define CDF families. These include higher order forms of stochastic dominance, which allow CDFs in a stochastic dominance relation to cross. Higher order forms of stochastic dominance relax (3) by placing the inequality condition on integrals of CDFs rather than directly on CDFs, or on integrals of integrals of CDFs, etc. Nth order stochastic dominance has been found useful for n up to 3. In this paper we deal only with first order stochastic dominance, the

most important. Since the bounding CDFs of our example stand in a first order stochastic dominance relation, the body of existing work on stochastic dominance can be drawn upon for methods of using the derived bounding CDFs. The theory and application of stochastic dominance is fairly well developed, with at least one book [22] and hundreds of papers. Stochastic dominance has been most extensively investigated in economics and finance as a basis for optimization and decision making. Stochastic dominance has also been applied to computer systems analysis [20]. A survey of recent work and many references appear in Levy [21].

The mathematics field of majorization theory is closely related to stochastic dominance.

Another way to define a family of CDFs is useful in answering questions like, "What actual but unknown CDFs are consistent with a set of measurements?" Some general results were developed by Kolmogorov and Smirnov and summarized by Kolmogorov in 1941 [23], and are easily restated using interval terminology. Such results, as well as the present work, help indicate some significant advantages of CDFs over PDFs in representing probabilistic information.

8 Discussion

Let us review some promising applications, then compare the automatically verified histogram method with the better known Monte Carlo methods.

8.1 Applications

An important next step in the development of the automatically verified histogram method is to apply it to interesting problems. A simulation application similar to the disk overloading problem is described in great detail by Berleant et al. [24]. More complex simulation problems are a natural extension. One simulation problem is described briefly next, after which two other areas of application are mentioned.

Consider the problem of forecasting the spread of the disease AIDS (acquired immune deficiency syndrome) among intravenous drug abusers due

to sharing of needles. Rate of spread has been modeled as

$$\frac{dN(t)}{dt} = cN(t)^v - \mu N(t) \tag{4}$$

where $N(t)$ is the number of intravenous drug abusers, c is a constant factor, v is a constant exponent, and μ is the constant rate at which individuals leave the population. The exponent v is believed to be in the interval $[0, 1)$. Yet there seems no good reason to consider any particular distribution of probability mass over that interval a better description of our knowledge of the value of v than many other distributions [25]. Therefore v is best described with an interval. However $N(t)$ is known well enough to be described by a PDF [26]. Hence simulation modeling of this problem [27] appears to be a good candidate for the automatically verified histogram method.

The fields of economics and finance are also natural candidates for applying the automatically verified histogram method, as stochastic dominance has been best developed in those fields.

The conventional histogram method has been applied mostly in reliability analyses. Thus reliability analyses and decision analyses in which reliability plays an important role also form a promising application area for the verified histogram method.

8.2 Monte Carlo methods

The automatically verified histogram method forms an interesting comparison with the well known Monte Carlo approach. Table 1 summarizes.

8.2.1 Comparative disadvantages of the automatically verified histogram method

Monte Carlo methods may be applied to models with dependent inputs, if those dependencies are well characterized. The automatically verified histogram method cannot be, although if the dependencies are well characterized, a modification of the automatically verified histogram method in accordance with Section 4.1 should be feasible.

	Automatically verified histogram method	Histogram discretization method	Monte Carlo methods
Handles dependent inputs?	×	×	√
Automatically verifying?	√ (excess width likely)	×	× (insufficient width almost certain)
Handles interval inputs?	√	√	√
Handles PDF inputs?	√	√	√
Handles PDFs and intervals?	√	√	×

Table 1: Comparison of Monte Carlo and histogram methods.

8.2.2 Comparative advantages of the automatically verified histogram method

With Monte Carlo methods, input values are chosen randomly for each input variable to generate a vector of input values. This input vector generation process is done numerous times to sample the space of possible input vectors in a statistically adequate way. Each input vector is applied to the model, which produces the corresponding output vector. If inputs are interval valued, the range of values that are *observed* for a particular output variable over the set of input vectors is used to describe the range of values that are *possible* for that output variable — a process that inherently produces unguaranteed results. Thus we must be satisfied with some notion of statistical adequacy that falls short of a guarantee.

If the inputs to a model are PDFs instead of intervals, the space of possible input vectors can be randomly sampled in such a way that samples

are generated consistently with the PDFs describing the inputs. The set of output values produced for a given output variable can be statistically analyzed to describe its PDF, or better its CDF (Kolmogorov 1941) [23], although results are still not guaranteed.

Monte Carlo methods have difficulty with situations in which some input variables are intervals and others are PDFs, due to the difficulty of adequately sampling an input space consisting of both intervals and PDFs.

Thus there are two main advantages of the automatically verified histogram method over the Monte Carlo approach:

- *The automatically verified histogram method produces guaranteed results, unlike Monte Carlo methods.*
- *The automatically verified histogram method appears better suited to a mixture of interval and PDF valued inputs than Monte Carlo methods.*

9 Acknowledgements

J. Chang, H. Cheng, J. Conrad, and A. Maqsood read drafts of this paper. Software for follow on work is currently under development by H. Cheng and A. Maqsood, based on an existing implementation of the histogram discretization method written for personal computers by K. Bognæs [28].

The author thanks the reviewers, especially number one, for helpful comments and suggestions.

References

[1] Evans, R. A. *Bayes paradox.* IEEE Transactions on Reliability **R–31** (4) (1982), p. 321.

[2] Ingram, G. E., Welker, E. L., and Herrmann, C. R. *Designing for reliability based on probabilistic modeling using remote access computer systems.* In: "Proceedings 7th reliability and maintainability conference", American Society of Mechanical Engineers, 1968, pp. 492–500.

[3] Colombo, A. G. and Jaarsma, R. J. *A powerful numerical method to combine random variables.* IEEE Transactions on Reliability **R–29** (2) (1980), pp. 126–129.

[4] Jackson, P. S., Hockenbury, R. W., and Yeater, M. L. *Uncertainty analysis of system reliability and availability assessment.* Nuclear Engineering and Design **68** (1981), pp. 5–29.

[5] Ahmed, S., Clark, R. E., and Metcalf, D. R. *A method for propagating uncertainty in probabilistic risk assessment.* Nuclear Technology **59** (1982), pp. 238–245.

[6] Corsi, F. *Mathematical models for marginal reliability analysis.* Microelectronics and Reliability **23** (6) (1983), pp. 1087–1102.

[7] Rushdi, A. M. and Kafrawy, K. F. *Uncertainty propagation in fault-tree analyses using an exact method of moments.* Microelectronics and Reliability **28** (6) (1988), pp. 945–965.

[8] Kaplan, S. *On the method of discrete probability distributions in risk and reliability calculations — application to seismic risk assessment.* Risk Analysis **1** (3) (1981), pp. 189–196.

[9] Moore, R. E. *Risk analysis without Monte Carlo methods.* Freiburger Intervall-Berichte 1 (1984), pp. 1–48.

[10] Moore, A. S. *Interval risk analysis of real estate investment: a non-Monte Carlo approach.* Freiburger Intervall-Berichte 3 (1985), pp. 23–49.

[11] Neumaier, A. *Interval methods for systems of equations.* Cambridge University Press, 1990.

[12] Moore, R. E. *Methods and applications of interval analysis.* SIAM, 1979.

[13] Moore, R. E. *On computing the range of a rational function of n variables over a bounded region.* Computing **16** (1976), pp. 1–15.

[14] Asaithambi, N. S., Zuhe, S., and Moore, R. E. *On computing the range of values.* Computing **28** (1982), pp. 225–237.

[15] Cornelius, H. and Lohner, R. *Computing the range of values of real functions with accuracy higher than second order.* Computing **33** (1984), pp. 331–347.

[16] Alefeld, G. *Enclosure methods.* In: Ullrich, C. (ed.) "Computer arithmetic and self-validating numerical methods", Academic Press, 1990, pp. 55–72.

[17] Kolev, L. V., Mladenov, V. M., and Vladov, S. S. *Interval mathematics algorithms for tolerance analysis.* IEEE Transactions on Circuits and Systems **35** (8) (1988), pp. 967–975.

[18] Hyvönen, E. *Constraint reasoning based on interval arithmetic: the tolerance propagation approach.* Artificial Intelligence **58** (1992), pp. 71–112.

[19] Berleant, D. and Kuipers, B. *Qualitative-numeric simulation with Q3.* In: Faltings, B. and Struss, S. "Recent advances in qualitative physics", MIT Press, Cambridge, Massachusetts, 1992, pp. 3–16.

[20] Post, G. V. and Diltz, J. D. *A stochastic dominance approach to risk analysis of computer systems.* Management Science Quarterly **10** (1986), pp. 363–375.

[21] Levy, H. *Stochastic dominance and expected utility: survey and analysis.* Management Science **38** (4) (1992), pp. 555–593.

[22] Whitmore, G. A. and Findlay, M. C. (eds.) *Stochastic dominance: an approach to decision-making under risk.* Lexington Books, Lexington, Massachusetts, 1978.

[23] Kolmogoroff, A. (*a.k.a.* Kolmogorov) *Confidence limits for an unknown distribution function.* Annals of Mathematical Statistics **12** (4) (1941), pp. 461–463.

[24] Berleant, D., Chandra, C., Bognæs, K., Liaw, C., Sheng, L., and Ch'ng, J. *Probabilities of qualitative behaviors for dependability analysis of a fault tolerance model.* Conference Proceedings, Symposium on Applied Computing, ACM Press, New York, 1992, pp. 883–889.

[25] Caulkins, J. P. and Kaplan, E. H. *AIDS impact on the number of intravenous drug users.* Interfaces **21** (3) (1991), pp. 50–63.

[26] *National household survey on drug abuse: population estimates 1991.* National Institute on Drug Abuse. 5600 Fishers Lane, Rockville, Maryland 20857, 1991.

[27] Berleant, D., Goforth, R. R., and Yuan, J. *A computer model for predicting AIDS among intravenous drug abusers.* "Proceedings of the Arkansas Academy of Science", Monticello, Arkansas, 1993.

[28] Bognæs, K. A. *Using probability distribution functions to manage poorly quantified data in interactive simulations.* Master's thesis, Department of Computer Systems Engineering, University of Arkansas, Fayetteville, Arkansas, 1993.

Dept. of Computer Systems Engineering
University of Arkansas
Fayetteville, AR 72701
USA
E-mail: djb@engr.engr.uark.edu