Grid-based Histogram Arithmetic for the Probabilistic Analysis of Functions [⋆]

Carlos Carreras[+] and Manuel V. Hermenegildo[⋆]

[+] *Department of Electrical Engineering*
[⋆] *Department of Computer Science*
Technical University of Madrid
28040 Madrid, SPAIN
[+] carreras@die.upm.es, [⋆] herme@fi.upm.es

Abstract. The selection of predefined analytic grids (partitions of the numeric ranges) to represent input and output functions as histograms has been proposed as a mechanism of approximation in order to control the tradeoff between accuracy and computation times in several areas ranging from simulation to constraint solving. In particular, the application of interval methods for probabilistic function characterization has been shown to have advantages over other methods based on the simulation of random samples. However, standard interval arithmetic has always been used for the computation steps. In this paper, we introduce an alternative approximate arithmetic aimed at controlling the cost of the interval operations. Its distinctive feature is that grids are taken into account by the operators. We apply the technique in the context of probability density functions in order to improve the accuracy of the probability estimates. Results show that this approach has advantages over existing approaches in some particular situations, although computation times tend to increase significantly when analyzing large functions.

Keywords: Interval computations, probabilistic analysis, estimation, approximate arithmetic, abstract interpretation.

1 Introduction

Recently, there has been increasing interest and activity in the theory and application of Interval Analysis and Interval Computation [1, 12, 14]. These techniques are recognized as a powerful tool for manipulating imprecise data and dealing with uncertainty. Therefore, they provide a formal basis for abstractions aimed to support quantitative approximation processes in a large number of application areas ranging from, e.g., robotics to constraint programming [2, 3, 11].

The point of view of using interval arithmetic as an *abstraction* can be described formally in terms of *abstract interpretation* [8]. A set of values in the

[⋆] The authors would like to thank the anonymous reviewers for their comments on previous versions of this paper. This work was funded in part by projects CICYT TIC97-0928 and TIC99-1151.

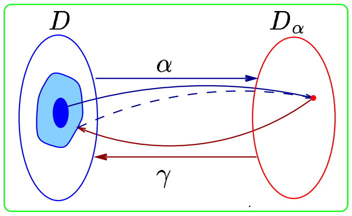

Fig. 1. Relationship between domains in abstract interpretation

concrete domain D of operands (generally a numeric domain, either continuous or discrete but typically large) is approximated by a set of intervals. Each such set of intervals can be seen as an element of a non-standard domain D_α, called an *abstract* domain, which is usually a complete lattice. D_α is then the set that contains all the admissible sets of intervals. We consider two monotonic mappings (i.e., mappings f which satisfy $x \subseteq y \;\Rightarrow\; f(x) \subseteq f(y)$) which relate the concrete and abstract domains and which are called *abstraction* $\alpha : D \mapsto D_\alpha$, and *concretization* $\gamma : D_\alpha \mapsto D$ (see Figure 1). Given a set of values v in D the *abstraction* function $\alpha(v)$ returns the corresponding (minimal) set of intervals. Conversely, given a set of intervals (an element i of D_α), the *concretization* function $\gamma(i)$ returns a (possibly infinite) set of concrete values from D.

Also, for each component operation *op* which operates on elements of D (e.g., $+$, $*$,...) an *abstract counterpart* op_α ($+_\alpha$, $*_\alpha$,...) is defined that operates on the corresponding sets of intervals in D_α. These abstract operations $+_\alpha$, $*_\alpha$,... are the standard interval arithmetic operations, augmented to operate on sets of intervals. A function is then computed or approximated by replacing the operators in the program by their abstract counterparts and applying the resulting *abstract function* to sets of inputs at a time, such sets being represented as sets of intervals. In order to reason about the correctness of this process, partial order relations are considered in the concrete and abstract domains: $\langle D, \subseteq \rangle$ and $\langle D_\alpha, \sqsubseteq \rangle$. The definition of $\sqsubseteq$ is induced by $\subseteq$ (set inclusion in D) and α such that $\forall i, i' \in D_\alpha : \; i \sqsubseteq i' \Leftrightarrow \gamma(i) \subseteq \gamma(i')$, i.e., a set of intervals i is "smaller" than another set of intervals i' if it corresponds to fewer values in D.

The standard interval operations ($+_\alpha$, $*_\alpha$, ...) do verify two important properties. The first one is that they *compute safe approximations*, i.e., given two sets of concrete values a and b, then $a+b \subseteq \gamma(\alpha(a) +_\alpha \alpha(b))$ (where by $a+b$ we mean the set of results of pairwise adding all elements of a and b), $a * b \subseteq \gamma(\alpha(a) *_\alpha \alpha(b))$, etc. I.e., it is guaranteed that the intervals which are result of an operation contain all possible values that can be obtained from the operation of values from the interval operands. However, these operations are not completely *precise* in the sense that if there are data dependencies between the operands (e.g., due to variables appearing more than once in the computation), the interval result is not guaranteed to be the minimum interval that contains all possible output values (i.e. data dependencies are only considered at the interval level, not at the level of individual values) [12, 9].

Regarding the second property, consider representing the intervals corresponding to the operands of abstract operations as sets of disjoint subintervals, i.e., consider a new abstraction function α' which represents an operand a instead of by $i = \alpha(a)$ by another element $i' = \alpha'(a)$ $(i, i' \in D_\alpha)$ composed of disjoint subintervals of i (note that then $\gamma(i) = \gamma(i')$). Now consider generalized versions of the interval operations $+_{\alpha'}$, $*_{\alpha'}$,... which operate such sets by computing their Cartesian product and merge the results in a single interval. Then $\alpha(a) +_{\alpha'} \alpha(b) \sqsubseteq \alpha(a) +_\alpha \alpha(b)$, $\alpha(a) *_{\alpha'} \alpha(b) \sqsubseteq \alpha(a) *_\alpha \alpha(b)$, etc. I.e., the results of the operations on subintervals are included in the intervals obtained from operating the original interval operands: the accuracy of the output can increase. This is because there is less loss of precision in the abstract operations due to the fact that data dependencies are considered in greater detail [12, 14]. In the limit, assigning one subinterval to each possible value of each operand would yield an exact interval result in all cases, e.g., $a * b = \gamma(\alpha_{limit}(a) *_\alpha \alpha_{limit}(b))$.

The two properties above allow defining abstractions based on predefined partitioning strategies, which will be referred to as *grids*. Such grids are analytic partitions of the numeric ranges of interest that force a specific representation, and are the basis for the definitions of new abstract domains and abstraction functions. Furthermore, it is possible to associate a value (e.g., a probability) with each element of the abstract domain associated with a given grid, resulting in the notion of *histogram grids* and the definitions of interval operations on such grids. This allows the application of this class of abstractions to probabilistic characterization of functions.

The application of interval methods to perform operations on probability density functions (PDFs) represented as histograms has been previously suggested (for detailed references see [4]). The standard interval data type is extended with a probability mass distributed inside the interval to form a histogram bar (called interval or bar in the following). This model does not provide information about the probability distributions inside the bars so only verifiable bounds on the output cumulative distribution functions (CDFs) can be obtained [4]. A common approximation to avoid this problem and obtain estimates of the output PFDs is to assume that the distributions inside the bars are always uniform [7]. This is justified by the second property above as long as discrete values are used (we consider integers) because, in the limit, the distribution of a probability mass inside an interval containing a single value is uniform.

The uniformity approximation allows using interval analysis for probabilistic (quantitative) function characterization. In fact, interval methods have been described as having more advantages than traditional random sampling approaches (i.e. Monte Carlo simulation) [10, 4]. Exhaustive exploration of the input data space is possible when represented in terms of intervals but, in general, it is infeasible when considering individual numeric values. Therefore, from the first property above, interval computations can provide safe bounds for the output distributions, while Monte Carlo approaches (based on partial random simulation of the input data space) cannot guarantee that the worst case scenario is actually considered in the results.

Two problems appear when applying intervals to PDF estimation. First, the computation of the Cartesian product of input histogram bars yields a set of output bars that must be merged into a single output histogram. However, the complexity of this merging can increase to infeasible levels (merging two intervals with non-empty intersection produces three smaller intervals, so every new merge is bound to deal with more intersections as computation progresses). Second, it has been argued that assuming uniform distributions inside the histogram bars can be a problem with some operations that significantly increase the size of the output interval but causing sparse distributions (i.e. integer multiplication).

Approaches based on the definition of grids have been proposed to minimize these problems and, in general, to control the accuracy of the enclosures (and PDFs) obtained through interval computations [7, 5, 6]. Histogram grids by forcing a specific representation on input and output histograms, allow controlling the sizes of their bars. However, the computation is typically still performed in terms of standard interval arithmetic.

In this paper, we introduce an alternative arithmetic (i.e., alternative abstract operations) to evaluate the effect of taking grids into consideration also during the computation. This arithmetic directly produces the histogram representation of an interval result in terms of the same grid used to represent the input intervals. This approach provides a more accurate probabilistic description of the operation result and thus allows increased accuracy in the output PDFs.

In the following sections, the notion of abstraction using grids is introduced as well as the corresponding abstract interval operations. Then, the notion of interval is generalized to that of a histogram bar, and the notion of histogram grids is presented, applied to the particular case of PDF computation. Then a new arithmetic, with operators based on a specific grid, and its computation model are presented. Finally, the new approach is compared to the case of performing computation using standard interval arithmetic (in terms of accuracy and computation times) when applied to a simple sequence of computations including data dependencies. Finally, the main conclusions are summarized.

2 Using Interval Grids as Abstractions

The definition of the abstract domain D_α is based on the so-called grids which are abstractions based on intervals.

Definition 1. *An interval $[a, b]$ is the set of $N = (b - a + 1)$ integers x that verify $a \leq x \leq b$.*

Definition 2. *A grid G is a partition of the concrete domain D in terms of intervals I_i: $G = \{I_i | \cup_{\forall i} I_i = D, \cap_{\forall i} I_i = \emptyset\}$.*

Grids can be defined by hand by the user or described through analytic models. Here, we consider analytic grids parameterized by a type, which determines their formal description, and a granularity (g), which determines the size of its intervals. In particular, the following two types are considered.

Definition 3. *A linear grid with granularity g is the set of adjacent intervals $[A, B]$, each of them uniquely identified by integer n, that verify one of the following identities:*

$$[A,B] = \begin{cases} [gn+1, g(n+1)] & (n < -1) \\ [-g+1, -1] & (n = -1) \\ [0,0] & (n = 0) \\ [1, g-1] & (n = 1) \\ [g(n-1), gn-1] & (n > 1) \end{cases}$$

Definition 4. *A geometric grid with granularity g is the set of adjacent intervals $[A, B]$, each of them uniquely identified by integer n, that verify one of the following identities:*

$$[A,B] = \begin{cases} \left[-g^{-n}+1, -g^{-(n+1)}\right] & (n < 0) \\ [0,0] & (n = 0) \\ \left[g^{(n-1)}, g^n - 1\right] & (n > 0) \end{cases}$$

In both definitions, integer n is called the *level* of the corresponding interval in the grid. In linear grids intervals are of equal size (except around the center), while in geometric grids interval size increases exponentially away from the origin. More complex grid models can be found in [5] where the center of symmetry of the grid can be moved from the origin to any other value.

Let's consider the set I of all possible intervals of D. In this situation, the set

$$I_G = \{i | i \in I, i \leq j, j \in G\}$$

where $\leq$ represents interval inclusion, is the set of all possible intervals allowed by a grid G, and the abstract domain induced by a grid G, $D_{\alpha,G}$ is defined by 2^{I_G}, i.e., it contains all the sets of possible intervals allowed by the grid. Given a set of concrete values V, an abstraction function can be associated with the grid which returns the abstract value corresponding to V in $D_{\alpha,G}$.

Definition 5 ($\alpha_G(V)$)**.** *The abstraction function associated with a grid G, $\alpha_G(V)$ is defined as:*

$$\alpha_G(V) = \{V_\alpha | V_\alpha \in D_{\alpha,G}, \forall v \in V, \exists^* V_\alpha / v \in V_\alpha \ \nexists v'_\alpha \in V_\alpha, v'_\alpha \sqsubseteq v_\alpha\}$$

This means that all concrete values in the same grid interval are represented by a single element of the abstract domain. For example, the set of integers $\{0, 1, 3, 4, 6\} \subset D$ is represented in terms of a linear grid with $g = 4$ as the set $\{[0,0], [1,3], [4,6]\} \subset D_{\alpha,lin(4)}$, or in terms of a geometric grid with $g = 2$ as the set $\{[0,0], [1,1], [3,3], [4,6]\} \subset D_{\alpha,geo(2)}$.

Standard definitions of operations between intervals are used as abstract operations for the computation [12, 14]. For example, in the case of positive intervals (those with both endpoints > 0):

$$[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]$$
$$[x_1, x_2] - [y_1, y_2] = [x_1 - y_2, x_2 - y_1]$$
$$[x_1, x_2] \times [y_1, y_2] = [x_1 \times y_1, x_2 \times y_2]$$
$$[x_1, x_2] \,/\, [y_1, y_2] = [x_1/y_2, x_2/y_1]$$

As an example of how the use of grids can improve accuracy consider the operation $[2, 4] * [8, 9]$ which results using standard interval arithmetic in $[16, 36]$. If a geometric grid with $g = 2$ is applied the interval $[2, 4]$ becomes $[2, 3], [4, 4]$ and thus the operation yields $[16, 27], [32, 36]$.

It should be noted that linear grids are better suited for sequences of additions and subtractions while geometric grids allow large reductions in the size of the input space and compensate for the range expansion produced by multiplications and exponentiations, at the cost of coarser intervals away from the origin.

3 Using Interval Histogram Grids as Abstractions

We now generalize the notion of interval by associating a weight with each such interval. In particular, and given the intended application to PDF computation, probabilities are assigned to intervals:

Definition 6. *An interval $[a, b]/p$ is the set of $N = (b - a + 1)$ integers x that verify $a \leq x \leq b$ with an associated probability mass p.*

It is assumed that p is uniformly distributed in $[a, b]$, so that the probability of any $x \in [a, b]$ can be computed as p/N. In this situation, a histogram is simply described as a set (ordered list) of disjoint generalized intervals. This assumption allows simplifying the computation model. The impact depends on the type of grid and the granularity selected. In the limit, if each interval contains a single integer value, probabilities are indeed uniform.

The grid-based approach for PDF estimation was partially introduced in [7] and later developed in [5, 6]. In particular, the representation of a generic histogram in terms of a given grid is governed by two rules:

- *Merge* rule: all intervals of the histogram occurring inside the same interval of the grid are represented as a single interval with probabilities added.
- *Split* rule: any interval of the histogram spanning over several intervals of the grid is decomposed into as many intervals with proportional probabilities before applying the merge rule.

These two rules are the key to controlling the number of bars in a histogram through the appropriate selection of a grid. They can be used to reduce the impact of the problems outlined in the previous section: the merging process that occurs when collecting output intervals in global histograms, and the uniformity assumption in large intervals obtained after operations causing sparse output distributions. Besides, they provide a formal mechanism to control the size of the interval input space (the Cartesian product of input bars) and, consequently, the estimation time.

Although this interval method of representation may suggest some resemblance to Latin Hypercube Sampling (LHS) as used in approaches based on Monte Carlo simulation, they are not related in any way as the interval method is based on a different computation model with different data types (i.e. intervals). (LHS divides the range of each of the k input variables into n non-overlapping intervals, randomly selects n values -one value from each interval- for each of the k variables, and combines them randomly into n k-tuplets which are used as input vectors for the simulation. While LHS reduces the number of samples for a given accuracy, they are much harder to compute so, in general, it has only a limited advantage with respect to standard Monte Carlo sampling [13, 15]).

Once grids have been selected for input and output representation, the histogram computation model is adapted from [4] as:

1. Consider the input space $N_1 \times \ldots \times N_I$ where N_i is the set of intervals describing the histogram of input i in terms of a selected grid.
2. For each vector $(\ldots, [a_{ij}, b_{ij}]/p_{ij}, \ldots)$ of the input space, where $[a_{ij}, b_{ij}]/p_{ij}$ represents the j-th bar of the histogram describing input i:
   (a) Compute its probability $P = \prod_{i=1}^{I} p_{ij}$.
   (b) Execute the operations using interval arithmetic.
   (c) Assign P to each resulting interval.
   (d) Collect the results in output histograms described in terms of selected grids applying the split and merge rules.

When considering a sequence of arithmetic operations, the use of grids allows controlling the size of the intervals in each input vector $(\ldots, [a_{ij}, b_{ij}]/p_{ij}, \ldots)$. However, the size of the interval(s) obtained after the computation (before applying the output grid) is determined by the type of operations in the sequence. Therefore, the approximation of assuming uniform distributions inside the intervals worsens if large intervals representing sparse distributions are obtained. The impact of this uniformity approximation is controlled through the value of P which is indirectly determined by the grid (finer grids reduce the impact but require longer computation times). However, it remains to be seen if using grid-based operators can provide advantages over using more detailed grids.

4 Grid-based Histogram Arithmetic

In the following sections, alternative arithmetic operators based on a specific grid are derived. The objective is to evaluate their impact in the estimation process. In particular, a geometric grid with $g = 2$, called G in the following, is considered. The choice of a geometric grid is simply due to the fact that it is more interesting and novel than a linear one. A granularity value of 2 is the most appropriate for the ranges of values being considered in the examples. Larger granularities are useful for larger value ranges. A generalization of these models for other grids is out of the scope of this study, although it should be fairly easy in the case of geometric grids.

Assuming interval operands represented according to G the new operators return the result of the operation also represented in terms of G. When this result includes more than one interval (in general, it is a histogram), the operator distributes the probability P among the output intervals according to the behavior of the specific arithmetic operation. (In order to have consistent input and output data types, this new arithmetic can be formulated as histogram arithmetic, as inputs can be viewed as histograms having a single interval).

From the previous description, the input of any intermediate operation in any run of the computation is a histogram. In this situation, the computation model of the previous section is modified as:

1. Consider the input space $N_1 \times \ldots \times N_I$ where N_i is the set of intervals describing the histogram of input i in terms of G.
2. For each vector $(\ldots, [a_{ij}, b_{ij}]/p_{ij}, \ldots)$ of the input space:
   (a) Compute the probability $P = \prod_{i=1}^{I} p_{ij}$.
   (b) Transform each $[a_{ij}, b_{ij}]/p_{ij}$ into a histogram with a single bar $[a_{ij}, b_{ij}]/1$.
   (c) For each operation with input histograms H_k (with one or several intervals), and for each combination of intervals from the Cartesian product of the intervals of histograms H_k::
      i. Represent each interval as a single-bar histogram.
      ii. Obtain a histogram result using G-based histogram arithmetic.
      iii. Proceed with the next operation, if any (step c). Then return.
      iv. Multiply the probability of each resulting histogram interval by P.
      v. Collect the result histogram in an output histogram in terms of G.

This computation model is much more complex because a tree of histograms is generated from each input vector $(\ldots, [a_{ij}, b_{ij}]/p_{ij}, \ldots)$. Of course, there is a risk of a combinatorial explosion in the number of histograms produced, and grid-based operators should be carefully used. The theorems presented in the following sections provide hints on the conditions to avoid this problem by setting bounds on the number of intervals generated by the operations.

4.1 Addition/Subtraction Model

Subtractions are treated as particular cases of additions by considering that if $[a, b]$ is an interval at level l, $-[a, b] = [-b, -a]$ is at level $-l$.

Two interval operands $[a_1, a_2]/p_a$ and $[b_1, b_2]/p_b$ at levels l_a and l_b of G (geometric grid with $g = 2$), containing N_a and N_b integers respectively, are considered. When they are added, the endpoints of the output range are computed as $[c_1, c_2] = [a_1 + b_1, a_2 + b_2]$ (from standard interval arithmetic).

Theorem 1. The addition of two intervals A and B at positive levels l_a and l_b of G produces at most two intervals at levels $\max\{l_a, l_b\}$ and $(\max\{l_a, l_b\} + 1)$.

Proof. Considering the largest intervals at levels l_a and l_b and the endpoints at levels $\max\{l_a, l_b\}$ and $(\max\{l_a, l_b\} + 1)$,

$$A + B = [2^{l_a - 1} + 2^{l_b - 1}, 2^{l_a} + 2^{l_b} - 2] \subset [2^{max\{l_a, l_b\} - 1}, 2^{max\{l_a, l_b\} + 1} - 1]$$

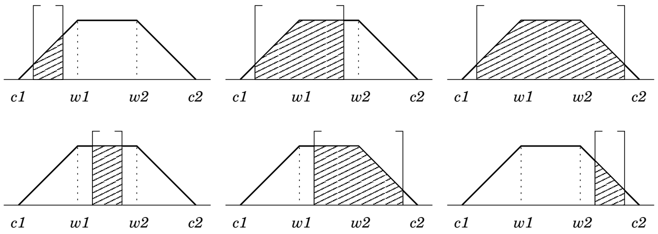

Fig. 2. Relative positions between a grid interval and the distribution of results in the addition of intervals with uniform density functions

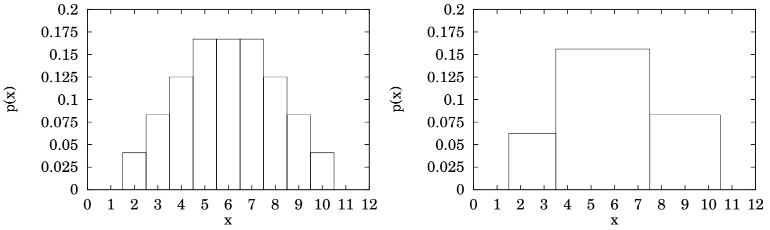

Fig. 3. Exact and G-based density functions of $[-7,-4]/1 + [9,14]/1$

Assuming uniform distributions in the operands, the distribution of the $M = N_a N_b$ integer results or occurrences in $[c_1, c_2]$ has the general shape of a trapezoid, with a height $h = \min\{N_a, N_b\}$, and corners c_1, c_2, $w_1 = (c_1 + h - 1)$, and $w_2 = (c_2 - h + 1)$ (see Figure 2). The number of occurrences m of any value $x \in [c_1, c_2]$ can be obtained as:

$$m = \begin{cases} x - c_1 + 1 & c_1 \leq x < w_1 \\ h & w_1 \leq x \leq w_2 \\ c_2 - x + 1 & w_2 < x \leq c_2 \end{cases}$$

As $[c_1, c_2]$ must be described in terms of G, in general, it becomes a set of intervals (histogram) with probabilities proportional to the previous distribution of occurrences. These probabilities are obtained from analyzing the possible positions of a grid interval with respect to the three sections of the distribution above (Figure 2).

Definition 7. *The G-based addition of two intervals $[a_1, a_2]/p_a$ and $[b_1, b_2]/p_b$ with N_a and N_b integers respectively, produces the set of intervals described by*

$$\begin{cases} [c_1, c_2]\,/p_a p_b & \text{if} \quad l_1 = l_2 \\ \left[c_1, 2^{l_1} - 1\right]/p_{l_1}, \bigcup_{i=l_1+1}^{l_2-1}[2^{i-1}, 2^i - 1]/p_i, \left[2^{l_2-1}, c_2\right] p_{l_2} & \text{if} \quad l_1 \neq l_2 \end{cases}$$

where l_1 is the level including $c_1 = a_1 + b_1$, and l_2 is the level including $c_2 = a_2 + b_2$. When $l_1 \neq l_2$, for each interval $[x_1, x_2]/p_x$ of the set, $p_x = p_a p_b M_x / N_a N_b$

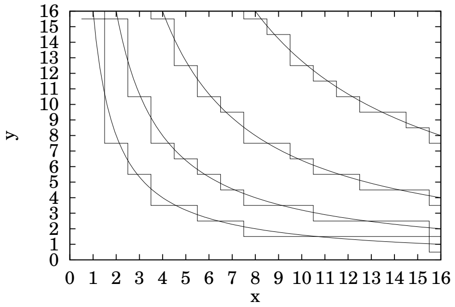

Fig. 4. Area approximation

with M_x obtained as

$$M_x = \begin{cases} (x_1 + x_2 - 2c_1 + 2)(x_2 - x_1 + 1)/2 & x_2 < w_1 \\ (w_1 + x_1 - 2c_1 + 1)(w_1 - x_1)/2 + (x_2 - w_1 + 1)h & x_1 < w_1, w_1 \leq x_2 \leq w_2 \\ (w_1 + x_1 - 2c_1 + 1)(w_1 - x_1)/2 + (w_2 - w_1 + 1)h + & \\ \quad +(2c_2 - w_2 - x_2 + 1)(x_2 - w_2)/2 & x_1 < w_1, x_2 > w_2 \\ h(x_2 - x_1 + 1) & w_1 \leq x_1 \leq w_2, x_2 \leq w_2 \\ (w_2 - x_1 + 1)h + (2c_2 - w_2 - x_2 + 1)(x_2 - w_2)/2 & w_1 \leq x_1 \leq w_2, x_2 > w_2 \\ (2c_2 - x_1 - x_2 + 2)(x_2 - x_1 + 1)/2 & x_1 > w_2 \end{cases}$$

As an example, Figure 3 represents the output distribution of the addition $[-7, -4]/1 + [9, 14]/1$. The plot on the left is the exact density function. The plot on the right is obtained with the G-based operator. It should be noted that if standard interval arithmetic is applied, a uniform distribution (at $p(x) = 0.111$) in $[2, 10]$ is obtained.

4.2 Multiplication and Division Models

Multiplication produces, in general, sparse distributions of results in wide ranges. So interval results contain values that cannot be obtained from the corresponding integer multiplication. (Only positive intervals are considered here as sign computation can be performed independently).

Theorem 2. The product of two intervals A and B at levels l_a and l_b of G, produces at most two intervals at levels $(l_a + l_b - 1)$ and $(l_a + l_b)$.
Proof. Considering the largest intervals at levels l_a and l_b and the endpoints at levels $(l_a + l_b - 1)$ and $(l_a + l_b)$, and applying standard interval arithmetic

$$A \times B = [2^{l_a+l_b-2}, 2^{l_a+l_b} - 2^{l_a} - 2^{l_b} + 1] \subset [2^{l_a+l_b-2}, 2^{l_a+l_b} - 1]$$

The model of the G-based interval multiplication is based on a computation in the concrete domain. Considering the region defined by $x \in [a_{1r}, a_{2r}] = [a_1 - 0.5, a_2 + 0.5]$ and $y \in [b_{1r}, b_{2r}] = [b_1 - 0.5, b_2 + 0.5]$, the number of products below

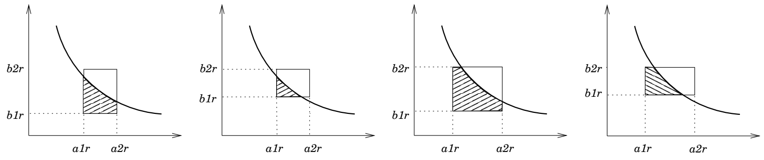

Fig. 5. Relative positions of $xy = K$ in the defined region

a value K can be approximated by the area below the curve $xy = K$ included in the previous region. From the previous theorem, a value $K = 2^{l_a+l_b-1}$ separates the products that belong to each of the two output intervals. An approximation of the number of occurrences in the lower output interval can be obtained as:

$$A = \int_{x_1}^{x_2} \left(\frac{K}{x}\right) dx - C = K \ln\left(\frac{x_2}{x_1}\right) - C$$

As shown by some examples in Figure 4, it is an approximation because this expression provides the real area below the curve, instead of the number of unit squares corresponding to integer occurrences. (A more sophisticated model accounting for long tails that do not include unit squares is used in the implementation to reduce the impact of the approximation). The values of x_1, x_2 and C are obtained from analyzing the possible positions of the curve $xy = K$ with respect to the rectangle defined by the ranges of x and y. These positions are represented in Figure 5.

Definition 8. *The G-based multiplication of two intervals, $[a_1, a_2]/p_a$ and $[b_1, b_2]/p_b$ at levels l_a and l_b and with N_a and N_b integers respectively, produces the intervals*

$$\begin{cases} [c_1, 2^{l_a+l_b-1} - 1]/p, [2^{l_a+l_b-1}, c_2]/p_a p_b - p \text{ if } & c_1 < 2^{l_a+l_b-1} \leq c_2 \\ [c_1, c_2]/p_a p_b & \text{else} \end{cases}$$

where $c_1 = a_1 b_1$, $c_2 = a_2 b_2$, $K = 2^{l_a+l_b-1}$, $p = \left(\frac{p_a p_b}{N_a N_b}\right)\left(K \ln\left(\frac{x_2}{x_1}\right) - C\right)$, and

$$x_1 = \begin{cases} a_{1r} & K \leq a_{1r} b_{2r} \\ K/b_{2r} & K > a_{1r} b_{2r} \end{cases}$$
$$x_2 = \begin{cases} a_{2r} & K > a_{2r} b_{1r} \\ K/b_{1r} & K \leq a_{2r} b_{1r} \end{cases}$$
$$C = \begin{cases} b_{1r}(x_2 - x_1) & K \leq a_{1r} b_{2r} \\ b_{1r}(x_2 - x_1) - N_b(x_1 - a_{1r}) & K > a_{1r} b_{2r} \end{cases}$$

The model for division is based on similar ideas. In this case, it is assumed that the endpoints of the denominator cannot be 0.

Theorem 3. The division of two intervals A and B at levels l_a and l_b of G, produces at most two intervals at levels $(l_a - l_b)$ and $(l_a - l_b + 1)$.

Proof. Considering the largest intervals at levels l_a and l_b and the endpoints at levels $(l_a - l_b)$ and $(l_a - l_b + 1)$, and applying standard interval arithmetic

$$A/B = [2^{l_a-1}/2^{l_b} - 1, 2^{l_a} - 1/2^{l_b-1}] \subset [2^{l_a-l_b-1}, 2^{l_a-l_b+1}]$$

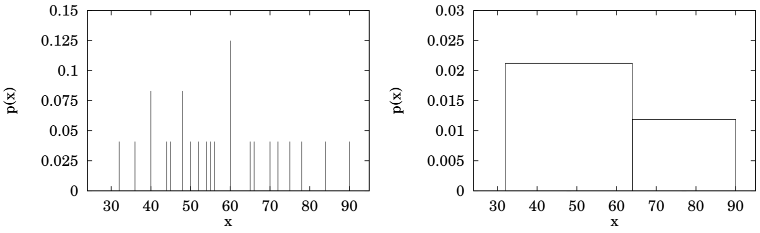

Fig. 6. Exact and G-based density functions of $[4,6]/1 \times [8,15]/1$

The curve to be considered in this case for the area computation is $x/y = K$ with $K = 2^{l_a - l_b}$, so

$$A = \int_{x_1}^{x_2} \left(\frac{x}{K}\right) dx - C = \frac{x_2^2 - x_1^2}{2K} - C \tag{1}$$

Definition 9. *The G-based division of two intervals, $[a_1, a_2]/p_a$ and $[b_1, b_2]/p_b$ at levels l_a and l_b and with N_a and N_b integers respectively, produces the intervals*

$$\begin{cases} [0,0]/p_a p_b & \text{if} \quad l_a < l_b \\ \left[c_1, 2^{l_a - l_b} - 1\right]/p_a p_b - p,\ \left[2^{l_a - l_b}, c_2\right]/p & \text{if} \quad l_a \geq l_b, c_1 < 2^{l_a - l_b} \leq c_2 \\ [c_1, c_2]/p_a p_b & \text{else} \end{cases}$$

where $c_1 = a_1/b_2$, $c_2 = a_2/b_1$, $K = 2^{l_a - l_b}$, $p = \left(\frac{p_a p_b}{N_a N_b}\right)\left(\frac{x_2^2 - x_1^2}{2K} - C\right)$, and

$$x_1 = \begin{cases} a_{1r} & K \leq a_{1r}/b_{1r} \\ b_{1r}K & K > a_{1r}/b_{1r} \end{cases}$$
$$x_2 = \begin{cases} a_{2r} & K > a_{2r}/b_{2r} \\ b_{2r}K & K \leq a_{2r}/b_{2r} \end{cases}$$
$$C = \begin{cases} b_{1r}(x_2 - x_1) & K > a_{2r}/b_{2r} \\ b_{1r}(x_2 - x_1) - N_b(a_{2r} - x_2) & K \leq a_{2r}/b_{2r} \end{cases}$$

As an example, exact and G-based plots are represented in Figure 6 for the multiplication $[4,6]/1 \times [8,15]/1$. The uniform distribution obtained from standard interval arithmetic has a constant density at $p(x) = 0.017$.

5 Example of a Computation

Although it is clear that individual G-based operators are more accurate than standard interval operators, it is also important to characterize their behavior when considering sequences of operations (implying data dependencies). For this purpose, the following example of computation is considered (from the reliability estimation of a robot arm joint [6]):

$$j = s_a \times s_b + m - s_a \times s_b \times m$$

where s_a and s_b are sensor probabilities of failure described by the left plot of Figure 7, and m is a motor probability of failure represented in the plot on the right of Figure 7.

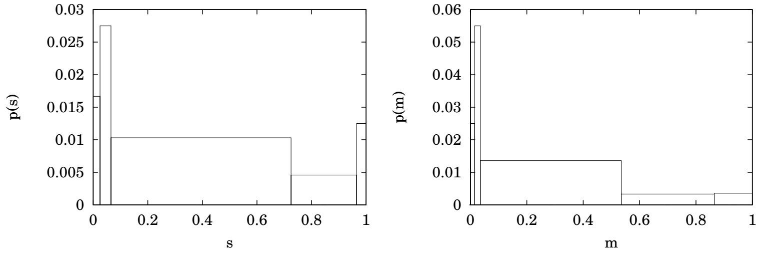

Fig. 7. Probability density functions of s_a and s_b (left), and m (right)

It can be observed that in order to apply the estimation approaches presented here it is required to scale the data (and the computation) for a description in the integer domain. In particular, probabilities of failure with two fractional digits are considered, thus requiring a scaling by 100. For more details, see [6]. (Such scaling is undone in the representation of the computation results).

The plots in figures 8 to 11 represent the density functions of j obtained from four different computational approaches. In each figure, the plot on the left corresponds to the result of the approach represented by merging exactly all the individual results (in general, the intersection of two individual results produces three output bars). The plot on the right is the representation of the same result when an output grid G (geometric with $g = 2$) is used to collects the results.

Figure 8 displays the exact output density functions obtained from the exhaustive exploration of the 10^6 vectors of the input space (considering integers). Note that the "small" size of the problem allows obtaining this exact result, in general unknown, and that the peak around 0.5 cannot be totally represented when using G.

The plots in Figure 9 are obtained by using standard interval computations on the intervals of histograms in Figure 7. Again, collection times, as previously explained, do not become prohibitive due to the small size of the problem. In this case, data dependencies are taken into account at a coarse level of detail (large input intervals), so the approximation is poor and the results merely show the peak around 0.5.

The results obtained applying an input grid G with standard interval arithmetic are represented in Figure 10. The approximation is much better than without grids. The two peaks of the PDF are clearly seen. However, when represented in terms of G, probability masses do not distribute as in the exact PDF (the second peak appears displaced around the value 0.25).

Finally, Figure 11 contains the plots computed using G-based arithmetic, which also implies an input grid G. Once again, results are improved, as both

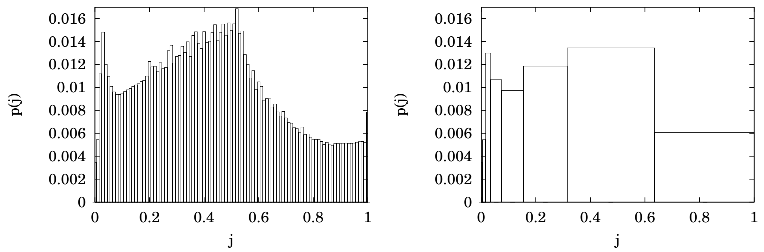

Fig. 8. Exact PDF of j

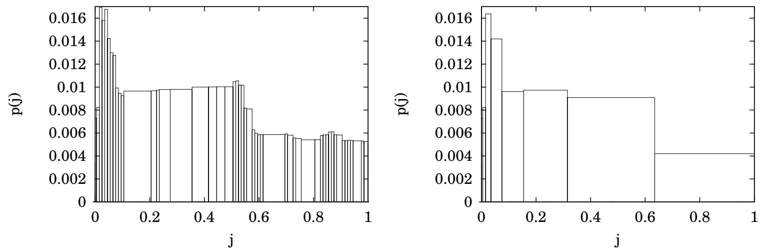

Fig. 9. PDF of j computed with intervals but without grids

peaks are reflected in the plots but, in this case, the second peak appears around 0.4, so the representation in terms of G is the best of the three approximations.

Table 1 contains some statistics from the four computations, including the number of operations performed (Size), the computation time in msec. (Time), the number of intervals saved to disk after the computation (Memory. This number also includes intervals describing the inputs and intermediate variables), and the error in the result. This error is obtained by comparing the representations in terms of G. It is a weighted percentage of the exact distribution computed as

$$\mathrm{Error}(\%) = 100 \times \sum_{\forall \mathrm{bars}} \frac{\mid \mathrm{Exact} - \mathrm{Approx} \mid}{2 \times \mathrm{Exact}}$$

where Exact is the probability of a bar in the exact distribution and Approx is the probability of the corresponding bar in the approximate distribution. The factor 2 accounts for the fact that each misplaced results causes a difference in the distributions of twice its probability.

The table shows several interesting results. Clearly, exhaustive integer exploration of the input space (first two rows) would be infeasible in larger examples, as the computation time is a function of the input space size. The impact of collection times can be observed by comparing results with and without an output grid. Using no output grid causes a significant increase in the total time even though this is a small example with a limited number of output values (100).

Errors confirm the qualitative analysis of the plots previously made. The improvement achieved with G-based operators is at the cost of longer (but accept-

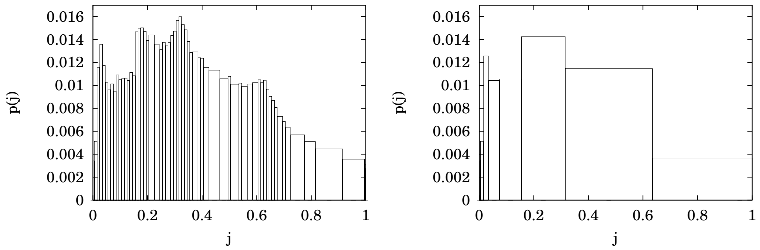

Fig. 10. PDF of j computed with an input grid G and standard interval arithmetic

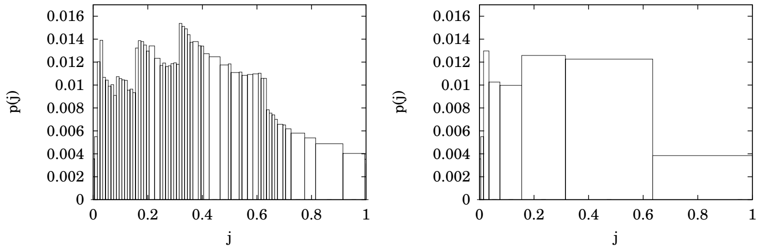

Fig. 11. PDF of j computed with grid-based arithmetic (input grid G is implied)

able) computation times. As previously mentioned, this cost can increase significantly in larger examples as the computation with G-based operators can generate many more intermediate results than standard interval operators. However, the theorems from the previous section seem to anticipate moderately longer computation times when using operators based on grids like the one used here (geometric with $g = 2$), as most interval operations are proven to generate at most two output intervals.

6 Conclusions

In this paper, after introducing grids as abstractions with the objective of improving the precision of interval computations, a new set of approximate arithmetic operators for probabilistic characterization of functions has been presented. The new operators bring grids into the behavior of the operators themselves. Results from operators developed for a particular grid show that this approach provides the ability to control the accuracy and computation times of the estimation process at a different level than approaches based on grids for input and output representation. The new approach reduces the error of the PDF estimates at the cost of longer computation times. The results from a particular example show that this is a moderate increase, although this can be different in other examples with different grids and data sets.

Input	Computation	Output	Size	Time	Memory	Error (%)
integer	integer	no grid	6181806	2773459	20844	-
integer	integer	geometric(2)	6181806	2297749	10651	0
no grid	no grid	no grid	750	490	739	-
no grid	no grid	geometric(2)	750	240	602	17.42
geometric(2)	no grid	no grid	3072	1640	1159	-
geometric(2)	no grid	geometric (2)	3072	820	642	6.55
geometric(2)	geometric(2)	no grid	3072	2120	1490	-
geometric(2)	geometric(2)	geometric(2)	3072	1260	642	3.94

Table 1. Statistics from the Computation

References

1. G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* AP, NY, 1983.
2. APIC'95. International Workshop on Applications of Interval Computations, El Paso, Texas, February 1995.
3. Frédéric Benhamou, David McAllester, and Pascal Van Hentenryck. CLP(Intervals) Revisited. In *Proceedings of ILPS'94*, pages 1–21, Ithaca, NY, USA, 1994. MIT Press.
4. D. Berleant. Automatically Verified Reasoning with Both Intervals and Probability Density Functions. *Interval Computations*, 1993(2):48–70, 1993.
5. C. Carreras, J.A. López, and O. Nieto-Taladriz. Bit-width Selection in Datapath Implementations. In *Proc. 12th IEEE International Symposium on System Synthesis*, pages 114–119, San Jose, CA, Nov 1999.
6. C. Carreras and I.D. Walker. Interval Methods for Improved Robot Reliability Estimation. In *Proc. IEEE Annual Reliability and Maintainability Symposium, RAMS 2000, Los Angeles, CA*, Jan 2000.
7. C. Carreras, I.D. Walker, O. Nieto-Taladriz, and J.R. Cavallaro. Robot Reliability Estimation Using Interval Methods. In *Proc. MISC'99 International Workshop on Applications of Interval Analysis to Systems and Control*, pages 371–385, Girona, Spain, Feb 1999.
8. P. Cousot and R. Cousot. Abstract Interpretation: a Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Fourth ACM Symposium on Principles of Programming Languages*, pages 238–252, 1977.
9. Eero Hyvonen. Evaluation of Cascaded Interval Functions. In *Proceedings of Intervational Workshop on Constrain-Based Reasoning, 8th Florida AI Research Symposium*, April 1995.
10. Janne Pesonen et al. Interval Approach Challenges Monte Carlo Simulation. In *Proceedings of Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN-95)*, 1995.
11. Kim Marriot and Peter Stuckey. *Programming with Constraints: An Introduction.* The MIT Press, 1998.
12. R.E. Moore. *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, 1979.

13. W.H. Press, S.A. Teukolsky, W.T.Vetterling, and B.P. Flannery. *Numerical Recipes in FORTRAN: The Art of Scientific Computing.* Cambridge Univ. Press, New York, 1992.
14. H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions.* Ellis-Horwood, Chichester, 1988.
15. C.N. Zeeb and P.J. Burns. A Comparison of Failure Probability Estimates by Monte Carlo Sampling and Latin HyperCube Sampling, 1998.