Exact Bounds for Interval Functions Under Monotonicity Constraints, with Potential Applications to Paleontology

Emil Platon
Energy & Geoscience Institute
University of Utah
423 Wakara Way, Suite 300
Salt Lake City, UT 84108, USA
eplaton@egi.utah.edu

Vladik Kreinovich, Scott A. Starks
Pan-American Center for Earth and Environmental Studies
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@cs.utep.edu

Karen Villaverde
Dept. of Computer Science
New Mexico State University
Las Cruces, NM, 88003, USA
kvillave@cs.nmsu.edu

ABSTRACT

In paleontology, we know that in each area, the age of a fossil monotonically increases with depth. We have several observations of age and depth – both known with interval uncertainty – and we would like to find, for each possible depth, the interval of the possible values of the corresponding age. A similar problem of bounding an intervally defined function under monotonicity constraint occurs in many other application areas. In this paper, we provide an efficient algorithm for solving this problem.

Categories and Subject Descriptors

J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Earth and atmospheric sciences*; F.2.1 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Numerical Algorithms and Problems*

General Terms

Algorithms, Theory

1. CASE STUDY: PALEONTOLOGY

1.1 Paleontology is important

Crudely speaking, paleontology is a study of fossils. Such a study is extremely important for evolutionary biology, and it is also very important for other geological sciences: a fossil found in a geological stratum provides additional useful information for dating that stratum. A fossil also often gives us a good understanding of what was happening at this location at the corresponding time: for example, a coral is an unambiguous indication of a warm ocean.

1.2 The notion of a stratigraphic map

One way of determining the age of a fossil is based on the fact that as we go deeper and deeper, we find older and older fossils. So, in principle, we can use the depth x at which the fossil has been found to determine its age y.

The exact dependence between the depth x and the age y – called a *stratigraphic map* – is different for different locations, because it depends on the geological history of this location. At some past epochs, sediments were forming at a higher accumulation rate; for such epochs, the depth increases with time. At other epochs, the rate of accumulation was much slower, so we may have fossils of different ages at approximately the same depth.

1.3 Main ideas behind constructing a stratigraphic map

How is the stratigraphic map constructed now? In every area, we have several fossils whose age y has been determined. For each fossil type, we may have several fossils of this type, with different age. In this case, we select one of these fossils:

- in field studies, we select the oldest fossil as the most reliable, because the deepest fossils are the least disturbed by the consequent geological processes;
- in the wells results, we select the youngest fossil as the most reliable, because the fossils that are the closest to the surface are the least disturbed by drilling.

In both cases, for the selected fossil, we know the depth x_i at which it was found, and we know the estimated age y_i. Based on the points (x_i, y_i), we must find the desired dependence $y = f(x)$.

Since deeper layers are older, we should have a monotonic (increasing) dependence $y = f(x)$ for which $y_i = f(x_i)$. So, ideally, we should have a monotonic function that passes through all the points.

1.4 The practical construction of a stratigraphic map is not that easy

The conclusion about monotonicity is based on the idealized assumption that y_i is the age of the oldest (for wells, youngest) of many fossils of this type. For some types, we do have many fossils, so the oldest of these fossils represents a reasonable size sample and is, therefore, highly reliable. For other types of fossils, however, we may have only a few sample fossils of this type in a given area; for such types, the corresponding age y_i and depth x_i are not very accurate.

As a result of this inaccuracy, in practice, it is usually impossible to have a monotonic dependence that passes exactly through all the points (x_i, y_i): we may have $x_i < x_j$ while $y_i > y_j$.

1.5 Traditional approach

The traditional paleontological approach to this problem is, crudely speaking, as follows (see, e.g., [7]). Since few-sample data points do not fit to a monotonic curve, we select a threshold n_0 and only consider points (x_i, y_i) which came from samples of size $\neq n_0$.

Ideally, we should select the smallest possible n_0 for which the values can still fit into a monotonic curve, i.e., for which $x_i < x_j$ always implies $y_i \leq y_j$.

1.6 Room for improvement

In the traditional approach, after setting up a threshold, we:

- ignore all the points (x_i, y_i) with lower accuracy, and
- consider all the points with higher accuracy as exact, ignoring the fact that these points are not absolutely accurate.

In both cases, it is desirable to use the ignored information instead of simply ignoring it:

- if we take into consideration the inaccuracy of the data (x_i, y_i) based on which we have built the stratigraphic map, then we would be able to determine the accuracy of this map;
- if, in addition to the data points that fit into a monotonic curve, we take into consideration less accurate data points as well, we will be hopefully able to construct a more accurate stratigraphic map.

1.7 Interval uncertainty

How can we describe the data accuracy? Inaccuracy means that, e.g., for the age, the actual (unknown) age y_i of all the (yet uncovered) fossils of the given type is, in general, somewhat different from the estimate oldest age $\widetilde{y}_i$.

Ideally, we should know the set of possible values of the estimation error $\Delta y_i \stackrel{\text{def}}{=} \widetilde{y}_i - y_i$, and we should know the probabilities of different values from this set. However, to be able to determine these probabilities, we need to have a large number of data points, and when we have a lot of data points, the estimate is pretty accurate anyway. Therefore, in the important cases in which we want to know the accuracy, we cannot experimentally determine these probabilities.

At best, we can find a confidence interval based on the known properties of the extreme-value statistics (see, e.g., [8, 23]), or just elicit these intervals from the experts.

So, if we take uncertainty into consideration, then, for each fossil type i, instead of the exact values x_i and y_i, we know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ and $\mathbf{y}_i = [\underline{y}_i, \overline{y}_i]$ that contain these unknown values.

1.8 Towards the precise formulation of the problem

Based on all the fossils found in a given area, we know the n boxes $\mathbf{x}_i \times \mathbf{y}_i$ corresponding to different types of fossils. We know that the monotonic dependence $y = f(x)$ is such that $y_i = f(x_i)$ for some $(x_i, y_i) \in \mathbf{x}_i \times \mathbf{y}_i$.

Our objective is to find, for every depth x, the bounds of the possible values of age $y = f(x)$ for all the dependencies that are consistent with the given data. This is the problem that we will solve in this paper.

1.9 Other practical applications of the resulting mathematical problem

Before we find the bounds on $f(x)$, we must first check that our interval bounds are consistent, i.e., that there exists a monotonic function that is consistent with all the boxes. This subproblem has many applications outside paleontology.

Indeed, in many problems in science and engineering, we know that a physical quantity y depends on the physical quantity x, i.e., $y = f(x)$ for some function $f(x)$, and we want to check whether this dependence is monotonic.

In *spectral analysis*, chemical species are identified by locating local maxima of the spectra; see, e.g., [18, 19]. Thus, to identify the chemical species, we must identify intervals between local extrema, i.e., intervals of monotonicity.

In *radioastronomy*, sources of celestial radio emission and their subcomponents are identified by locating local maxima of the measured brightness of the radio sky. In other words, we are interested in the local maxima of the *brightness distribution*, i.e., of the function $y(x)$ that describes how the intensity y of the signal depends on the position x of the point from which we receive this signal. Thus, in radioastronomy, we must also identify the intervals of monotonicity.

Elementary particles are identified by locating local maxima of the experimental curves that describe (crudely speaking) the scattering intensity y as a function of energy x. Thus, in elementary particle physics, finding intervals of monotonicity is also important.

In *1-D landscape analysis*, e.g., different mountain slopes are different monotonicity intervals; see, e.g., [1, 2, 4].

In *financial analysis*, it is also important to find intervals of monotonicity because they correspond to growth or decline periods; see, e.g., [6]

In *clustering*, different 1-D clusters correspond to a multi-modal distribution, so clusters can be naturally described as combinations of monotonicity intervals separating local minima of the probability density function; see, e.g., [12, 15, 16].

Local maxima and minima are also used in the methods that accelerate the convergence of the measurement result to the real value of a physical variable, and thus allow the user to estimate this value without waiting for the oscillations to stop [14]. Thus, to accelerate convergence, we must also be able to efficiently find intervals of monotonicity.

Algorithms for solving this (sub)problem have been previously described in [13, 20, 21, 22].

1.10 Additional complexity

The additional complexity comes from the fact that, as we have mentioned, it is possible to have several different ages for the same depth. In mathematical terms, this means that the dependence $y = f(x)$ is not necessarily a monotonic function, it may be a *limit* of the graphs of monotonic functions in the sense of Hausdorff metric (see, e.g., [17]).

We are now ready for the exact definitions and for the formulation of the result.

2. PRECISE FORMULATION OF THE PROBLEM AND THE MAIN RESULT

Definition 1. By a *monotonic dependence* f, we mean the graph of a continuous mapping $m(s) = (m_1(s), m_2(s))$ from the real line $\mathbb{R}$ to the plane $\mathbb{R}^2$ for which $t < s$ implies that $m_1(t) \le m_1(s)$ and $m_2(t) \le m_2(s)$.

It is easy to see that if the graph f is the graph of a function, then this definition is equivalent to this function being (non-strictly) monotonically increasing. Not every monotonic dependence is a function: e.g., a "step-function" for which $y = 0$ for $x < 0$, $y = 1$ for $x = 1$, and $y \in [0, 1]$ for $x = 0$, is a monotonic dependence but not a function.

Definition 2. By a *box*, we mean a Cartesian product of two intervals. We say that a monotonic dependence f is *consistent* with a box $\mathbf{x} \times \mathbf{y}$ if the graph f contains a point from this box, i.e., if $f \cap (\mathbf{x} \times \mathbf{y}) \ne \emptyset$.

Definition 3. By *data* d, we mean a finite collection of boxes. We say that a monotonic dependence f is *consistent* with the data d – and denote it $Con(f, d)$ – if f is consistent with each of the corresponding boxes. We say that the data d is *consistent* if there exists a monotonic dependence f that is consistent with this data.

THEOREM 1. *The data d is consistent if and only if for every i and j, $\overline{x}_i < \underline{x}_j$ implies $\underline{y}_i \le \overline{y}_j$.*

For consistent data, our objective is, given the data $[\underline{x}_i, \overline{x}_i] \times [\underline{y}_i, \overline{y}_i]$ $(1 \le i \le n)$ and a real number x, to find the exact lower and upper bounds of the corresponding y over all the monotonic dependences that are consistent with this data:

$$\underline{f}(x) \stackrel{\text{def}}{=} \inf\{y \mid \exists f\, ((x, y) \in f \,\&\, Con(f, d))\}. \quad (1)$$

$$\underline{f}(x) \stackrel{\text{def}}{=} \inf\{y \mid \exists f\, ((x, y) \in f \,\&\, Con(f, d))\}. \quad (2)$$

THEOREM 2.

$$\underline{f}(x) = \max_{i : \overline{x}_i < x} \underline{y}_i; \quad \overline{f}(x) = \min_{j : x < \underline{x}_j} \overline{y}_j. \quad (3)$$

PROOF. Let us denote

$$\underline{F}(x) \stackrel{\text{def}}{=} \max_{i : \overline{x}_i < x} \underline{y}_i; \quad \overline{F}(x) \stackrel{\text{def}}{=} \min_{j : x < \underline{x}_j} \overline{y}_j. \quad (4)$$

Let us first show that for every monotonic dependence f that is consistent with the given data d and for every y for which $(x, y) \in f$, the value y is located between $\underline{F}(x)$ and $\overline{F}(x)$.

Indeed, for every i, since f is consistent with d, there exists a pair $(x_i, y_i) \in f \cap (\mathbf{x}_i \times \mathbf{y}_i)$. For this pair, $\underline{x}_i \le x_i \le \overline{x}_i$, so $\overline{x}_i < x$ implies that $x_i < x$. Since $(x, y) \in f$ and $(x_i, y_i) \in f$, by definition of a monotonic dependence, the inequality $x_i < x$ implies that $y_i \le y$. Since $\underline{y}_i \le y_i$, we thus conclude that $y \ge \underline{y}_i$.

Since y is larger than or equal to $\underline{y}_i$ for all i for which $\overline{x}_i < x$, it is therefore larger than or equal to the largest of such $\underline{y}_i$, i.e., that $y \ge \underline{F}(x)$. We can similarly prove that $y \le \overline{F}(x)$.

If $\overline{x}_i < \underline{x}_j$, then, for any x from the open interval $(\overline{x}_i, \underline{x}_j)$, we have $\overline{x}_i < x < \underline{x}_j$. We have proven that for every monotonic dependence that is consistent with the data d, we have $\underline{y}_i \le y \le \overline{y}_j$. So, if the data d is consistent, then $\overline{x}_i < \underline{x}_j$ indeed implies $\underline{y}_i \le \overline{y}_j$ – this is exactly the condition from Theorem 1.

Vice versa, if this condition is satisfied, then we always have $\underline{F}(x) \le \overline{F}(x)$.

To complete the proof of Theorems 1 and 2, it is therefore sufficient to prove that both piece-wise constant monotonic functions $\underline{F}(x)$ and $\overline{F}(x)$, when extended to step-wise monotonic continuous dependences, are consistent with the data d, i.e., that for every k, each of these functions is consistent with the k-th box $\mathbf{x}_k \times \mathbf{y}_k$. Without losing generality, let us prove it for $\underline{F}(x)$.

Indeed, for a piecewise-constant step dependence like $\underline{F}(x)$, at each point x, the range of possible values of y goes from $\underline{F}(x-0) \stackrel{\text{def}}{=} \lim\limits_{\varepsilon>0,\varepsilon\to 0} \underline{F}(x-\varepsilon)$ to $\underline{F}(x+0) \stackrel{\text{def}}{=} \lim\limits_{\varepsilon>0,\varepsilon\to 0} \underline{F}(x+\varepsilon)$. Due to monotonicity, when x goes from $\underline{x}_k$ to $\overline{x}_k$, possible values of y go from $\underline{F}(\underline{x}_k - 0)$ to $\underline{F}(\overline{x}_k + 0)$. (Since the graph $\underline{F}$ is a graph of a continuous mapping from the real line, it is connected, so all the values from the corresponding intervals are possible.) Therefore, to prove that this graph intersects with the box, it is sufficient to prove that one of the possible values of y also belongs to the y-interval $[\underline{y}_k, \overline{y}_k]$, i.e., that

$$[\underline{F}(\underline{x}_k - 0), \underline{F}(\overline{x}_k + 0)] \cap [\underline{y}_k, \overline{y}_k] \neq \emptyset. \quad (5)$$

The formula for the intersection of the two intervals is well known: $[a, b] \cap [a', b'] = [\max(a, a'), \min(b, b')]$. Thus, the two intervals have a non-empty intersection if and only if $\max(a, a') \leq \min(b.b')$, i.e., if and only if $a \leq b'$ and $a' \leq b$. In our case, we must prove that $\underline{F}(\underline{x}_k - 0) \leq \overline{y}_k$ and $\underline{y}_k \leq \underline{F}(\overline{x}_k + 0)$.

By definition of $\underline{F}(x)$ (formula (4), we have:

$$\underline{F}(\overline{x}_k + \varepsilon) = \max_{i: \overline{x}_i < \overline{x}_k + \varepsilon} \underline{y}_i. \quad (6)$$

Since $\varepsilon > 0$, the inequality $\overline{x}_i < \overline{x}_k + \varepsilon$ holds for $i = k$. Thus, $\underline{F}(\overline{x}_k + \varepsilon)$ is the largest of several values including $\underline{y}_k$. Hence, $\underline{F}(\overline{x}_k + \varepsilon) \geq \underline{y}_k$, and in the limit $\varepsilon \to 0$, we get the desired inequality $\underline{F}(\overline{x}_k + 0) \geq \underline{y}_k$.

Similarly,

$$\underline{F}(\underline{x}_k - \varepsilon) = \max_{i: \overline{x}_i < \underline{x}_k - \varepsilon} \underline{y}_i. \quad (7)$$

The inequality $\overline{x}_i < \underline{x}_k - \varepsilon$ implies that $\overline{x}_i < \underline{x}_k$. We already know that this new inequality, in its turn, implies that $\underline{y}_i \leq \overline{y}_k$. Since all the maximized values $\underline{y}_i$ do not exceed $\overline{y}_k$, the largest of these values, i.e., $\underline{F}(\underline{x}_k - \varepsilon)$, also cannot exceed $\overline{y}_k$. In the limit, we get $\underline{F}(\underline{x}_k - 0) \leq \overline{y}_k$.

Both inequalities have been proven, and so are Theorems 1 and 2. □

3. RESULTING EFFECTIVE ALGORITHMS

3.1 Algorithms for checking consistency

For checking consistency, Theorem 1 leads to exactly the same condition as emerged, for a slightly different problem, in [20]. We can therefore use algorithms from [20] to check consistency of our data as well.

If we simply check the condition from Theorem 1 for all $i = 1, \ldots, n$ and all $j = 1, \ldots, n$, then checking this condition would require $O(n^2)$ comparisons – i.e., $O(n^2)$ computational steps.

We can check this condition faster if we use the fact that this condition is equivalent to the following auxiliary property:

$$\text{For every } i, \text{ we have } \underline{y}_i \leq \min_{j: \underline{x}_j \geq \overline{x}_i} \overline{y}_j. \quad (8)$$

To check this condition, we can perform the following four-stage algorithm:

- First, we sort the values $\underline{x}_i$ into an increasing sequence – this requires $O(n \cdot \log(n))$ steps. We correspondingly re-order the values $\overline{x}_i$, $\underline{y}_i$, and $\overline{y}_i$. After this stage, we can assume that the values $\underline{x}_i$ are sorted:

$$\underline{x}_1 \leq \underline{x}_2 \leq \ldots \leq \underline{x}_n.$$

- Then, for every i from 1 to n, we compute the value $M_i \stackrel{\text{def}}{=} \min(\overline{y}_n, \overline{y}_{n-1}, \ldots, \overline{y}_i)$. Here, $M_n = \overline{y}_n$. If we already know M_i, then we can compute the previous value M_{i-1} by using a single operation $M_{i-1} = \min(M_i, \overline{y}_{i-1})$. Thus, computing all n values requires n computational steps.
- For each i from 1 to n, we can now use binary search (see, e.g., [5]) to find the integer $m(i)$ for which $\underline{x}_{m(i)-1} < \overline{x}_i \leq \underline{x}_{m(i)}$ (if such a value exists). Each binary search requires $\log(n)$ computational steps; thus, n such searches require $O(n \cdot \log(n))$ steps.
- Finally, for every i from 1 to n for which $m(i)$ exists, we check whether $\underline{y}_i \leq M_{m(i)}$:
  - if this inequality holds for all such i, then the measurement data is consistent with monotonicity;
  - otherwise, the function $f(x)$ cannot be monotonic.

  Each checking requires one comparison, so to check that this inequality holds for all i from 1 to n, we need n comparisons.

Overall, we thus need $O(n \cdot \log(n)) + O(n \cdot \log(n)) + O(n) + O(n) = O(n \cdot \log(n))$ steps.

For large n, we may want to further speed up computations if we have several processors working in parallel. All fours stages of the above algorithm can be parallelized by known techniques. In particular, Stage 1 is a particular case of a general *prefix-sum* problem, in which we must compute the values a_n, $a_n * a_{n-1}$, $a_n * a_{n-1} * a_{n-2}$, ..., for some associative operation $*$ (in our case, $* = \min$).

If we have a potentially unlimited number of processors, then we can do the following (see, e.g., [10], for the information on how to parallelize the corresponding stages):

- on Stage 1, we can sort the values $\underline{x}_i$ in time $O(\log(n))$;
- on Stage 2, we can compute the values M_i (i.e., solve the prefix-sum problem) in time $O(\log(n))$;
- on Stage 3, we can use n processors, each of which compute the corresponding value $m(i)$ in time $O(\log(n))$;
- finally, on Stage 4, we can use n processors, each of which checks the corresponding inequality in time $O(1)$.

As a result, we can check monotonicity in time

$$O(\log(n)) + O(\log(n)) + O(\log(n)) + O(1) = O(\log(n)).$$

If we have $p < n$ processors, then we can:

- on Stage 1, sort n values in time $O((n \cdot \log(n))/p + \log(n))$ [10];
- on Stage 2, compute the values M_i in time $O(n/p + \log(p))$ [3];
- on Stage 3, we subdivide n indices i between p processors, so each processor computes $m(i)$ for n/p indices i; computing each index requires $\log(n)$ time, so the overall time is $(n/p) \cdot \log(n) = O((n \cdot \log(n))/p)$;

- finally, on Stage 4, each of p processors checks the desired inequality for its n/p indices; this requires time $O(n/p)$.

Overall, we thus need time $O\left(\frac{n \cdot \log(n)}{p} + \log(p)\right)$.

3.2 Algorithms for constructing lower and upper bounds

The function $\underline{f}(x)$ as described by the formula (3) is piecewise constant; when x increases, the value of $\underline{f}(x)$ can only change if when $x = \overline{x}_i$ for some i.

Thus, the compute the corresponding values of $\underline{f}(x)$, it is sufficient to sort the upper endpoints $\overline{x}_i$ into the increasing sequence $\overline{x}_1 \le \overline{x}_2 \le \ldots \le \overline{x}_n$, and then to compute the corresponding values $m_i \stackrel{\rm def}{=} \max(\underline{y}_1, \ldots, \underline{y}_i)$.

Similarly to the previous algorithm, sorting requires $O(n \cdot \log(n))$ steps and computing m_i requires n steps, so overall, we need $O(n \cdot \log(n))$ steps to compute $\underline{f}(x)$. Similarly, we need $O(n \cdot \log(n))$ steps to compute $\overline{f}(x)$, so the overall computational complexity is $O(n \cdot \log(n))$.

If we have a potentially unlimited number of processors working in parallel, then sorting requires time $O(\log(n))$ and computing m_i also requires time $O(\log(n))$, so overall, we need time $O(\log(n))$. If we have $p < n$ processors, then we need time $O\left(\frac{n \cdot \log(n)}{p} + \log(p)\right)$.

Overall, the computation time for computing the bounds is asymptotically the same as the time for checking consistency.

4. FROM MONOTONICITY TO MORE COMPLEX CONSTRAINTS

In some practical problems, we know not only that the unknown dependence is monotonic, we also know that its rate of increase cannot be smaller than a certain value $c > 0$. For example, in paleontology, we may know that the accumulation rate cannot exceed a certain value a; then $\frac{dy}{dx} \ge c \stackrel{\rm def}{=} 1/a$.

In such situations, we face a slightly different problem: given the data d, check whether there is a dependence $y(x)$ that is consistent with the data and for which, $\frac{dy}{dx} \ge c$ for all points x.

The condition $\frac{dy}{dx} \ge c$ is equivalent to $\frac{dz}{dx} \ge 0$ for a new auxiliary variable $z \stackrel{\rm def}{=} y - c \cdot x$. In terms of (x, z), the original boxes becomes parallelograms: for $x_i = \underline{x}_i$, we have an interval $[\underline{z}_i^-, \overline{z}_i^-] \stackrel{\rm def}{=} [\underline{y}_i - c \cdot \underline{x}_i, \overline{y}_i - c \cdot \underline{x}_i]$; for $x_i = \overline{x}_i$, we have an interval $[\underline{z}_i^+, \overline{z}_i^+] \stackrel{\rm def}{=} [\underline{y}_i - c \cdot \overline{x}_i, \overline{y}_i - c \cdot \overline{x}_i]$. So, we can reformulate the original problem as follows: check whether there exists a monotonic dependence $g(x)$ $(= f(x) - c \cdot x)$ that is consistent with all the resulting parallelograms.

Here, for every monotonic dependence g that is consistent with the parallelograms, there exists a point $(x_i, z_i) \in g$ that is inside the parallelogram. Thus, for $(x, z) \in g$, $x < \underline{x}_i$ implies $x < x_i$, hence $z \le z_i \le \overline{z}_i^-$. Similarly, $\overline{x}_i < x$ implies $z \ge \underline{z}_i^+$. Thus, $\underline{g}(x) \le y \le \overline{g}(x)$, where

$$\underline{g}(x) = \max_{i:\overline{x}_i < x} \underline{z}_i^+; \quad \overline{g}(x) = \min_{j:x<\underline{x}_j} \overline{z}_j^-. \tag{9}$$

Consistency means that $\underline{g}(x) \le \overline{g}(x)$ for every x, i.e., that $\overline{x}_i < \underline{x}_j$ implies that $\underline{z}_i^+ \le \overline{z}_i^-$, i.e., substituting the expressions for z in terms of y that x, that $\underline{y}_i - c \cdot \overline{x}_i \le \overline{y}_j - c \cdot \underline{x}_j$, hence $\overline{y}_j - \underline{y}_i \ge c \cdot (\underline{x}_j - \overline{x}_i)$, which is equivalent to $c \le (\overline{y}_j - \underline{y}_i)/(\underline{x}_j - \overline{x}_i)$. Similarly to the monotonic case, one can prove that the above expressions $\underline{g}(x)$ and $\overline{g}(x)$ are indeed the exact bounds on possible values of $z = y - c \cdot x$; thus, $\underline{g}(x) + c \cdot x$ and $\overline{g}(x) + c \cdot x$ are the exact bounds for y.

A natural next question is: what are the possible values of dy/dx? For every data, we can consider all the "differentiable" functions (in the limit-motivated generalized sense, to allow step functions) that are consistent with all the boxes (i.e., whose graphs intersect with all the boxes). For a given interval $[a, b]$, for each of such functions f, we can take a connected interval hull $co(f'([a, b]))$ of the range of the derivative, and consider the intersection $F'([a, b])$ of these ranges over all such f.

If this intersection $F'([a, b])$ contains negative values, this means that every function that is consistent with the data is sometimes decreasing, so no monotonically increasing function is consistent with the data.

In general, the above arguments show, in effect, that the range $F'([a, b])$ is equal to $\{x \,|\, p \le x \le q\}$, where

$$p \stackrel{\rm def}{=} \min_{i,j:a \le \overline{x}_i \le \underline{x}_j \le b} \frac{\underline{y}_j - \overline{y}_i}{\underline{x}_j - \overline{x}_i}, \quad q \stackrel{\rm def}{=} \min_{i,j:a \le \overline{x}_i \le \underline{x}_j \le b} \frac{\overline{y}_j - \underline{y}_i}{\underline{x}_j - \overline{x}_i}.$$

(So, if $p > q$, the range is empty.) This formula provides a $O(n^2)$ time algorithm for computing the range.

Comment. A similar algorithm was proposed in [12], for the special case when we know the exact values of x_i.

5. FUTURE WORK

In some cases, we know not only the boxes $\mathbf{x}_i \times \mathbf{y}_i$, but also the probabilities of different values (x_i, y_i) from these boxes. For such cases, it is desirable to find not only the bounds on the stratigraphic map $f(x)$, but also the probabilities of different monotonic dependences within these bounds. In particular, it is desirable to come up with the most probable dependence among all dependences that are consistent with the given data.

6. ACKNOWLEDGMENTS

This work was supported by NASA grant NCC5-209, by USAF grant F49620-00-1-0365, by NSF grants EAR-0112968, EAR-0225670, and EIA-0321328, by Army Research Laboratories grant DATM-05-02-C-0046, and by the NIH grant 3T34GM008048-20S1.

The authors are thankful to Luc Longpré and to all the participants of the Geoinformatics meeting at the San Diego

Supercomputer Center (August 13–15, 2004) for valuable discussions.

7. REFERENCES

[1] M. S. Aguiar, A. C. R. Costa, and G. P. Dimuro, ICTM: an interval tessellation-based model for reliable topographic segmentation, *Numerical Algorithms* (to appear).

[2] M. S. Aguiar, G. P. Dimuro, A. C. R. Costa, R. K. S. Silva, and V. Kreinovich, HPC-ICTM: the interval categorizer tesselation-based model for high performance computing, *Proceedings of the Workshop on State-of-the-Art in Scientific Computing PARA'04*, Lyngby, Denmark, June 20–23, 2004.

[3] G. E. Blelloch, Prefix sums and their applications, In: J. H. Reif, *Synthesis of Parallel Algorithms*, Morgan Kaufmann, San Mateo, California, 1993, pp. 35–60.

[4] D. D. Coblentz, V. Kreinovich, B. S. Penn, and S. A. Starks, Towards Reliable Sub-Division of Geological Areas: Interval Approach, In: L. Reznik and V. Kreinovich (eds.), *Soft Computing in Measurements and Information Acquisition*, Springer-Verlag, Berlin-Heidelberg, 2003, pp. 223–233.

[5] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[6] G. J. Deboeck, K. Villaverde, and V. Kreinovich, Interval Methods for Presenting Performance of Financial Trading Systems, *Reliable Computing*, 1995, Supplement (Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, Febr. 23–25, 1995), pp. 67–70.

[7] T. D. Demchuk, E. Platon, R. J. Fitzimmons, and R. F. Waszczaak, Quantitative analyses of palynological data with correlation to depositional paleoenvironments and sequence stratigraphy, ofshore West Africa. In: *Proceedings of the 31st International Geological Congress*, Rio de Janeiro, Brazil, August 2000.

[8] J. Galambos, *The asymptotic Theory of Extreme Order Statistics*, Wiley, New York, 1987.

[9] I. J. Good and R. A. Gaskins, Density Estimation and Bump-Hunting by the Penalized Likelihood Method Exemplified by Scattering and Meteorite Data, *J. Amer. Stat. Soc.*, 1980, Vol. 75, pp. 42–56.

[10] J. Jájá, *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading, MA, 1992.

[11] V. Kreinovich, H. T. Nguyen, G. P. Dimuro, A. C. R. Costa, and B. R. C. Bedregal, A New Differential Formalism for Interval-Valued Functions and Its Potential Use in Detecting 1-D Landscape Features, *Proceedings of the International Conference on Information Technology InTech'03*, Chiang Mai, Thailand, December 17–19, 2003, pp. 491–498.

[12] V. Kreinovich, E. J. Pauwels, S. Ferson, and L. Ginzburg, A Feasible Algorithm for Locating Concave and Convex Zones of Interval Data and Its Use in Statistics-Based Clustering, *Numerical Algorithms* (to appear).

[13] J. Lorkowski and V. Kreinovich, If we measure a number, we get an interval. What if we measure a function or an operator?, *Reliable Computing*, 1996, Vol. 2, No. 3, pp. 287–298.

[14] K. Nickel, Interval Acceleration of Convergence, in: R. E. Moore (editor), *Reliability in Computing*, Academic Press, N.Y., 1988, pp. 151–169.

[15] E. J. Pauwels and G. Frederix, Image Segmentation by Nonparametric Clustering Based on the Kolmogorov-Smirnov Distance, In: D. Vernon (ed.), *Proceedings of the 6th European Conference on Computer Vision ECCV'2000, Dublin, Ireland, June 26–July 1, 2000, Part II*, Springer Lecture Notes in Computer Science, Vol. 1843, 2000, pp. 85–99.

[16] E. J. Pauwels, G. Frederix, and G. Caenaen, Image segmentation based on statistically principled clustering, *Proc. of 2001 International IEEE Conference on Image processing*, 2001, Vol. 3, pp. 66–69.

[17] B. Sendov, *Hausdorff Approximations*, Kluwer, Dordrecht, 1990.

[18] D. J. Thompson, Spectrum Estimation Techniques for Characterization and Development of WT4 Waveguide, *Bell Syst. Technical Journal*, Part 1, November 1977, Vol. 56, p. 1769; Part 2, December 1977, Vol. 57, p. 1983.

[19] D. J. Thompson, Spectrum Estimation and Harmonic Analysis, *Proceedings IEEE*, 1982, Vol. 70, pp. 1055–1096.

[20] K. Tupelly, *Checking if There Exists a Monotonic Function that is Consistent with the Measurement Results*, Master's Thesis, Department of Computer Science, University of Texas at El Paso, 2004.

[21] K. Villaverde and V. Kreinovich, A linear-time algorithm that locates local extrema of a function of one variable from interval measurement results, *Interval Computations*, 1993, No. 4, pp. 176-194.

[22] K. Villaverde and V. Kreinovich, Parallel algorithm that locates local extrema of a function of one variable from interval measurement results, *Reliable Computing*, 1995, Supplement (Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, Febr. 23–25, 1995), pp. 212–219.

[23] H. M. Wadsworth (ed.), *Handbook of Statistical Methods for Engineers and Scientists*, McGraw-Hill Publishing Co., New York, 1990.