For Complex Intervals, Exact Range Computation Is NP-Hard Even for Single Use Expressions (Even for the Product)

Martine Ceberio
Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
mceberio@cs.utep.edu,
vladik@utep.edu

Günter Mayer
Fachbereich Mathematik
Universität Rostock
Universitätsplatz 1
D-18051 Rostock, Germany
guenter.mayer@uni-rostock.de

ABSTRACT

One of the main problems of interval computations is to compute the range $\mathbf{y}$ of the given function $f(x_1, \ldots, x_n)$ under interval uncertainty. Interval computations started with the invention of straightforward interval computations, when we simply replace each elementary arithmetic operation in the code for f with the corresponding operation from interval arithmetic. In general, this technique only leads to an enclosure $\mathbf{Y} \supseteq \mathbf{y}$ for the desired range, but in the important case of single use expressions (SUE), in which each variable occurs only once, we get the exact range. Thus, for SUE expressions, there exists a feasible (polynomial-time) algorithm for computing the exact range.

We show that in the complex-valued case, computing the exact range is NP-hard even for SUE expressions. Moreover, it is NP-hard even for such simple expressions as the product $f(z_1, \ldots, z_n) = z_1 \cdot \ldots \cdot z_n$.

Categories and Subject Descriptors

F.2.1 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Numerical Algorithms and Problems*; G.1.0 [**Mathematics of Computing**]: Numerical Analysis—*Error analysis*; G.4 [**Mathematics of Computing**]: Mathematical Software—*Algorithm design and analysis*

1. INTRODUCTION

1.1 Interval computations are important

In many practical problems, we are interested in the value of a physical quantity y that is difficult or even impossible to measure directly. Since it is difficult to measure y *directly*, we then measure y *indirectly*, i.e., we measure the values of easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y in a known way $y = f(x_1, \ldots, x_n)$, and then we use the results $\widetilde{x}_i$ of measuring x_i to compute the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for the desired quantity y.

Measurements are never 100% accurate; as a result, the measured value $\widetilde{x}_i$ is, in general, different from the actual value x_i of the measured property: $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$. As a result, the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ differs from the actual value $y = f(x_1, \ldots, x_n)$ of the desired quantity – even when we know the exact algorithm for the dependence $y = f(x_1, \ldots, x_n)$ between x_i and y.

Traditionally in science and engineering, it is assumed that we know the probability of different values of measurement errors Δx_i. These probabilities are usually determined when we *calibrate* the measuring instrument used to measure x_i, i.e., when we compare the results of measuring with this instrument and the results of measuring with a much more accurate *standard* measuring instrument. However, in many real life situations, we do not know these probabilities:

- in state-of-the-art measurements, the instrument we use is the best available; in such situations, there is no better measuring instrument and so, calibration is not possible;

- in manufacturing, calibration is, in principle, possible, but its cost is often much higher than the cost of the sensor itself.

In such cases, instead of the probabilities, we only know the bounds Δ_i on the absolute value of the measurement error provided by the manufacturer of the measuring instrument. In this case, after we get the measurement result $\widetilde{x}_i$, the only information that we have about the (unknown) actual value x_i of the i-th measured quantity is that x_i belongs to the interval $\mathbf{x}_i \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. In this case, we must determine the range

$$\mathbf{y} \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}$$

of possible values of $y = f(x_1, \ldots, x_n)$. Computing this range is the main problem of *interval computations*; see, e.g., [3].

1.2 Interval arithmetic

Functions $f(x_1, x_2)$ that represent elementary arithmetic operations such as $+$, $-$, $\cdot$, etc., are monotonic in each of their variables. For such functions, the desired range can be computed by considering the appropriate endpoints of the inputs intervals. Specifically:

- the range for the sum $f(x_1, x_2) = x_1 + x_2$ is equal to $[\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$;
- the range for the difference $f(x_1, x_2) = x_1 - x_2$ is equal to $[\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2]$;
- the range for the product $f(x_1, x_2) = x_1 \cdot x_2$ is equal to $[\underline{y}, \overline{y}]$, where

$$\underline{y} = \min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_2 \cdot \overline{x}_2),$$

$$\overline{y} = \max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_2 \cdot \overline{x}_2).$$

- the range for the inverse $f(x_1) = 1/x_1$ is equal to $[1/\overline{x}_1, 1/\underline{x}_1]$ if $0 \notin [\underline{x}_1, \overline{x}_1]$.

Similarly, we can derive explicit formulas for the ranges of elementary functions. For example, for $f(x_1) = x_1^2$, the range is equal:

- to $[\underline{x}_1^2, \overline{x}_1^2]$ if $0 \le \underline{x}_1$;
- to $[\overline{x}_1^2, \underline{x}_1^2]$ if $\overline{x}_1 \le 0$;
- to $[0, \max(\underline{x}_1^2, \overline{x}_1^2)]$ if $\underline{x}_1 \le 0 \le \overline{x}_1$.

These formulas form *interval arithmetic*:

- the range of the sum is called the sum of the intervals,
- the range of the difference is called the difference between the intervals,
- the range of the square is called the square of the interval, etc.

1.3 General case: need for enclosure

In general, computing the exact range of the given function $f(x_1, \ldots, x_n)$ is an NP-hard problem – even for quadratic functions $f(x_1, \ldots, x_n)$; see, e.g., [5]. This means, in effect, that any algorithm for computing the range will need, in the worst case, computation time which grows exponentially with the number n of inputs. For large n, the resulting computation time becomes unrealistically long.

Since we cannot compute the exact range $\mathbf{y}$, we can therefore compute an *enclosure* $\mathbf{Y} \supseteq \mathbf{y}$ for this range.

1.4 Straightforward interval computations

One technique for computing such an enclosure, called *straightforward interval computations*, consists of the following:

- First, we *parse* the expression $f(x_1, \ldots, x_n)$, i.e., represent it as a sequence of elementary arithmetic operations. We do not have to invent new algorithms for this parsing: parsing is what compilers do anyway when they translate the code for computing $f(x_1, \ldots, x_n)$ into a sequence of hardware-supported elementary operations.
- Second, we replace each elementary arithmetic operation with the corresponding operation of interval arithmetic.

It is known that the resulting interval is always an enclosure for the desired range.

Sometimes, this enclosure coincides with the exact range. For example, to compute the range of the function $f(x_1) = \dfrac{1}{4} - \left(x_1 - \dfrac{1}{2}\right)^2$ over the interval $\mathbf{x}_1 = [0, 1]$, we first parse this expression into the following sequence of elementary operations:

$$r_1 = x_1 - \frac{1}{2}; \quad r_2 = r_1^2; \quad y = \frac{1}{4} - y_2,$$

and then replace each of these three operations with the corresponding operation from interval arithmetic:

$$\mathbf{r}_1 = \mathbf{x}_1 - \frac{1}{2} = [0, 1] - [0.5, 0.5] = [0 - 0.5, 1 - 0.5] = [-0.5, 0.5];$$

$$\mathbf{r}_2 = \mathbf{r}_1^2 = [0, \max((-0.5)^2, 0.5^2)] = [0, 0.25];$$

$$\mathbf{y} = \frac{1}{4} - \mathbf{r}_2 = [0.25, 0.25] - [0, 0.25] =$$

$$[0.25 - 0.25, 0.25 - 0] = [0, 0.25].$$

This is the exact range of this function $f(x_1)$.

However, in general, straightforward interval computations is a feasible procedure, while the problem the computing the exact range is NP-hard. This implies that in some situations, the resulting enclosure will not be exact. Indeed, we get *excess width* for the same function as above – if we first open the parentheses and simplify the expression into $f(x_1) = x_1 - x_1^2$. For this new expression, parsing leads to $r_1 = x_1^2$, $y = x_1 - r_1$, and thus, straightforward interval computations lead to

$$\mathbf{r}_1 = [0, 1]^2 = [0^2, 1^2] = [0, 1];$$

$$\mathbf{y} = \mathbf{x}_1 - \mathbf{r}_1 = [0, 1] - [0, 1] = [0 - 1, 1 - 0] = [-1, 1].$$

The resulting interval encloses the actual range $[0, 0.25]$ but is much wider than this range.

1.5 Single use expressions (SUE)

There is an important class of expressions for which straightforward interval computations lead to the exact range: *single use* expressions (SUE, for short), in which each variable occurs exactly once; see, e.g., [2, 3].

For example, in the expression $f(x_1) = \dfrac{1}{4} - \left(x_1 - \dfrac{1}{2}\right)^2$, the variable x_1 occurs exactly once – and, not surprisingly, straightforward interval computations result in the exact range. In an equivalent expressions $f(x_1) = x_1 - x_1^2$, the variable x_1 occurs twice – and we get excess width.

Because of this property of SUE expressions, we can compute their exact range in polynomial time – by using straightforward interval computations.

1.6 From real arithmetic to complex arithmetic

Many physical quantities are complex-valued, e.g., complex amplitude and impedance in electrical engineering (not to mention wave function in quantum mechanics). Due to measurement uncertainty, after measuring a value of such a physical quantity, we do not get its *exact* value, we only get a *set* of possible values of this quantity; see, e.g., [4, 6].

For a real-valued quantity, we can describe this uncertainty by providing an interval $[\underline{x}, \overline{x}]$ that is guaranteed to contain the (unknown) actual value of the measured quantity. A complex-valued quantity $z = x + \mathrm{i} \cdot y$ consists, in effect, of two real-valued quantities: the real part x and the imaginary part y. To describe the uncertainty with which we know x, it is therefore reasonable to use two intervals:

- an interval $\mathbf{x} = [\underline{x}, \overline{x}]$ that is guaranteed to contain the real part x, and
- an interval $\mathbf{y} = [\underline{y}, \overline{y}]$ that is guaranteed to contain the imaginary part y.

Such a pair of intervals forms a *complex interval*; it will be denoted by $\mathbf{z} = \mathbf{x} + \mathrm{i} \cdot \mathbf{y}$.

On a complex plane, possible complex values $z \in \mathbf{z}$ from a complex interval form a rectangle $[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}]$.

1.7 Processing complex intervals: precise formulation of the problem

Similarly to the real-valued case, it is often difficult (or even impossible) to directly measure the value of the desired complex quantity z. In such situations, we can do the following:

- we measure auxiliary quantities $z_1, \ldots, z_n$ (in general, also complex-valued) which are related to z by a known relation $z = f(z_1, \ldots, z_n)$, and
- we apply the algorithm f to the results $\widetilde{z}_n, \ldots, \widetilde{z}_n$ of direct measurements, and produce $\widetilde{z} = f(\widetilde{z}_1, \ldots, \widetilde{z}_n)$ as the estimate for the desired quantity z.

Since measurements are never 100% accurate, the actual values z_i are, in general, different from the measured values $\widetilde{z}_i$; hence, our estimate $\widetilde{z}$ is, in general, different from the actual value z of the desired quantity.

Once we know the complex intervals $\mathbf{z}_1, \ldots, \mathbf{z}_n$ that describe the uncertainty of each direct measurement, we would like to know which complex values z are possible values of the desired quantity. In precise terms, we are given:

- a computable function $f(z_1, \ldots, z_n)$ from complex numbers to complex numbers;
- n intervals $\mathbf{z}_1, \ldots, \mathbf{z}_n$, and
- a complex number z.

We want to find out whether z belongs to the range

$$\{f(z_1, \ldots, z_n) : z_1 \in \mathbf{z}_1, \ldots, z_n \in \mathbf{z}_n\}.$$

We will call this problem the *range computation problem for complex intervals.*

Comment. In the real-valued case, the range of a continuous function $f(x_1, \ldots, x_n)$ over the set $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ is itself an interval $\mathbf{y} = [\underline{y}, \overline{y}]$. Thus, to find out whether a given value y belongs to this interval, it is sufficient:

- to find the endpoints $\underline{y}$, $\overline{y}$ of the range interval, and
- to compare a given number y with these endpoints: y belongs to the desired range if and only if $\underline{y} \le y \le \overline{y}$.

In view of this fact, in the real-valued case, it was sufficient to compute the two endpoints of the range.

In the complex case, the range can have a complicated shape; so, it is no longer easy to characterize this shape and we have to consider the range computation problem in the above more complicated form.

1.8 Processing complex intervals: computational complexity of the general case

Real numbers are a particular case of complex intervals – when the imaginary part is 0. Similarly, a real interval $\mathbf{x} = [\underline{x}, \overline{x}]$ can be always viewed as a particular case of a complex interval – namely, an interval $\mathbf{z} = \mathbf{x} + \mathrm{i} \cdot [0, 0]$.

Thus, the problem of computing the range of a real-valued function under interval uncertainty is a particular case of the range computation problem for complex intervals. Since for real-valued intervals this problem is, in general, NP-hard, it is NP-hard for complex intervals as well.

1.9 Natural question: what is the computational complexity of processing complex intervals in the SUE case

For real-valued intervals, there are many important cases when there is a feasible (polynomial-time) algorithm for exactly computing the range; the simplest such case is the case of SUE expressions.

A natural question is: can we feasibly solve the problem of (exactly) computing for SUE expressions in the complex case as well? In this paper, we show that the answer to this question is negative: for complex intervals, the range computation problem is NP-hard even for SUE expressions. Moreover, we show that it is NP-hard for such simple expressions as the product of several fuzzy numbers, bilinear expressions, or the second population moment.

2. MAIN RESULTS

The simplest possible arithmetic operations are addition, subtraction, and multiplication.

If we only allow addition and subtraction (or even multiplication by a real-valued constant), then we end up with a general linear expression of the type $f(z_1, \ldots, z_n) = a_0 + a_1 \cdot z_1 + \ldots + a_n \cdot z_n$. For this expression, we can explicitly compute the range $\mathbf{z} = \mathbf{x} + \mathrm{i} \cdot \mathbf{y}$, where

$$\mathbf{x} = a_1 \cdot \mathbf{x}_1 + \ldots + a_n \cdot \mathbf{x}_n;$$

$$\mathbf{y} = a_1 \cdot \mathbf{y}_1 + \ldots + a_n \cdot \mathbf{y}_n.$$

Both intervals can be explicitly computed, and a given complex value $z = x + \mathrm{i} \cdot y$ is possible if and only if $x \in \mathbf{x}$ and $y \in \mathbf{y}$. So, for functions consisting only of such operations, there is a feasible (actually, linear-time) algorithm for computing the corresponding complex range exactly.

However, if instead of only allowing additions, we only allow multiplications, the problem becomes NP-hard:

THEOREM 1. *The problem of computing the exact range of the product $f(z_1, \ldots, z_n) = z_1 \cdot \ldots \cdot z_n$ of complex numbers under interval uncertainty is NP-hard.*

Comment. For reader's convenience, we placed all the proofs in the last section.

Since the product is a SUE expression, we get the following corollary:

THEOREM 2. *The problem of computing the exact range for complex intervals is NP-hard even for SUE expressions.*

The product of n numbers is a polynomial of order n. Maybe the problem is computationally simpler if we restrict ourselves to lower order SUE polynomials – e.g., quadratic or bilinear ones? Alas, no:

THEOREM 3. *The problem of computing the exact range of the scalar (dot) product $f(z_1, \ldots, z_n, t_1, \ldots, t_n) = z_1 \cdot t_1 + \ldots + z_n \cdot t_n$ under complex interval uncertainty is NP-hard.*

THEOREM 4. *The problem of computing the exact range of the second population moment $f(z_1, \ldots, z_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} z_i^2$ under complex interval uncertainty is NP-hard.*

3. AUXILIARY RESULTS

For the product, the problem remains NP-hard even if instead of computing the exact range, we simply want to compute the smallest box $\mathbf{x} \times \mathbf{y}$ that contains the actual range:

THEOREM 5. *The problem of computing the smallest box $\mathbf{x} \times \mathbf{y}$ that contains the range of the product $f(z_1, \ldots, z_n) = z_1 \cdot \ldots \cdot z_n$ of complex numbers under interval uncertainty is NP-hard.*

In the real-valued case, for SUE expressions, we can compute the smallest interval containing the actual range by using straightforward interval computations, i.e., by replacing each operation with numbers in the original formula f by the corresponding operation with intervals. For the product of complex numbers, not only we do not get the smallest box at the end, but the result of the corresponding operation-by-operation interval operations may actually depend on the order of multiplications – because for complex numbers, multiplication is, in general, not associative. For example, for $(1 - \mathrm{i}) \cdot (1 + \mathrm{i}) \cdot ([0, 1] - \mathrm{i})$, we get:

$$(1+\mathrm{i})\cdot([0,1]-\mathrm{i}) = ([0,1]+1)+\mathrm{i}\cdot([0,1]-1) = [1,2]+\mathrm{i}\cdot[-1,0];$$

$$(1-\mathrm{i})\cdot([1,2]+\mathrm{i}\cdot[-1,0]) = ([1,2]+[-1,0])+\mathrm{i}\cdot([-1,0]-[1,2]) = [0,2] + \mathrm{i} \cdot [-3,-1],$$

while $(1 - \mathrm{i}) \cdot (1 + \mathrm{i}) = 2$ hence

$$2 \cdot ([0,1] - \mathrm{i}) = [0,2] - 2\mathrm{i} \neq [0,2] + \mathrm{i} \cdot [-3,-1].$$

A similar result holds if instead of box-shaped complex intervals, we consider circular complex intervals $(\widetilde{z}, r) = \{z : |z - \widetilde{z}| \leq \Delta\}$. Here, it is natural to ask a similar question: given a function f, n circular intervals $\mathbf{z}_i$, and a complex number z, does z belong to the range $f(\mathbf{z}_1, \ldots, \mathbf{z}_n)$? It turns out that this range computation problem is NP-hard as well:

THEOREM 6. *The problem of computing the exact range of the scalar (dot) product $f(z_1, \ldots, z_n, t_1, \ldots, t_n) = z_1 \cdot t_1 + \ldots + z_n \cdot t_n$ under circular complex interval uncertainty is NP-hard.*

THEOREM 7. *The problem of computing the exact range of the second population moment $f(z_1, \ldots, z_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} z_i^2$ under circular complex interval uncertainty is NP-hard.*

4. PROOFS

Proof of Theorem 1. To prove NP-hardness of our range computation problem, we will reduce, to this new problem, a known NP-hard *partition problem*; see, e.g., [1, 5]. The subset problem is as follows: given n positive integers $s_1, \ldots, s_n$ and an integer s_0, to check whether there exists values $\varepsilon_i \in \{-1, 1\}$ such that $\sum_{i=1}^{m} \varepsilon_i \cdot s_i = 0$.

For every instance of the partition problem, we compute $k = 1/\left(\sum_{j=1}^{n} s_j\right)$, $\theta_i = k \cdot s_i$, $t_i = \tan(\theta_i)$, and we take $\mathbf{z}_i = 1 + \mathrm{i} \cdot [-t_i, t_i]$ and $z = \prod_{i=1}^{n} \sqrt{1 + t_i^2}$. Let us prove that this number z belongs to the range of the product if and only if the original instance of the partition problem has a solution.

In this proof, we will use the known fact that every complex number $z = x + \mathrm{i} \cdot y$ can be represented in a polar form $z = \rho \cdot e^{\mathrm{i} \cdot \alpha}$, where $\rho = \sqrt{x^2 + y^2}$ is the absolute value (magnitude) of z, and the "phase" θ is the angle between the direction from 0 to z and the positive real semi-axis. When we multiply complex numbers, their magnitudes multiply and their phases add.

Let us first prove that if the original instance has a solution ε_i, then z is equal to the product of n values $z_i = 1 + \mathrm{i} \cdot \varepsilon_i \cdot t_i \in \mathbf{z}_i$. Indeed, since $|z_i| = \sqrt{1 + t_i^2}$, the product of the magnitude is the desired value z. The angle α_i corresponding to each z_i is equal to $\alpha_i = \varepsilon_i \cdot \theta_i$, so the sum α of these angles is equal to $\sum_{i=1} \varepsilon_i \cdot \theta_i$. Since $\theta_i = k \cdot s_i$, we conclude that $\alpha = k \cdot \sum_{i=1}^{n} \varepsilon_i \cdot s_i$, i.e., $\alpha = 0$. So, this product $z_1 \cdot \ldots \cdot z_n$ has the right magnitude and the right angle and is, thus, equal to z.

Conversely, let us assume that z belongs to the range, i.e., that z can be represented as the product $z_1 \cdot \ldots \cdot z_n$ for some $z_i \in \mathbf{z}_i$. In other words, for this product, the magnitude is equal to z, and the phase α is 0. For each value $z_i = 1 + \mathrm{i} \cdot y_i \in \mathbf{z}_i$, its magnitude is equal to $\sqrt{1 + y_i^2}$. Since $|y_i| \leq t_i$, this magnitude cannot exceed $\sqrt{1 + t_i^2}$, and it is equal to $\sqrt{1 + t_i^2}$ only for the two endpoints $y_i = \pm t_i$.

If for some i, we have $y_i \in (-t_i, t_i)$, then the resulting magnitude is the product of several numbers all of which are $\leq \sqrt{1 + t_i^2}$ and some are smaller – thus, the magnitude of the product will be smaller than z. Since the magnitude of the product is equal to z, then, for each i, we have $z_i = 1 + \mathrm{i} \cdot \varepsilon_i \cdot t_i$ for some $\varepsilon_i \in \{-1, 1\}$. For each of these numbers z_i, the phase α_i is equal to $\varepsilon_i \cdot \theta_i$. Thus, from the fact that the overall angle $\alpha = \sum_{i=1}^{n} \alpha_i$ is equal to 0, we conclude that $\sum_{i=1}^{n} \varepsilon \cdot \theta_i = 0$, and, since $\theta_i = k \cdot s_i$, that $\sum_{i=1}^{n} \varepsilon_i \cdot s_i = 0$ – i.e., the original instance of the partition problem indeed has a solution. This completes the proof of the theorem.

Proof of Theorem 3. In this proof, we will use the reduction to the same partition problem as in Theorem 1. For every instance $s_1, \ldots, s_n$ of the partition problem, we take $\mathbf{z}_i = \mathbf{t}_i = \sqrt{s_i} \cdot (1 + \mathrm{i} \cdot [-1, 1])$ and $z = 0$. Then, possible values $z_i \in \mathbf{z}_i$ have the form $z_i = \sqrt{s_i} \cdot (1 + \mathrm{i} \cdot a_i)$ for some $a_i \in [-1, 1]$, and possible values of t_i are of the form $t_i = \sqrt{s_i} \cdot (1 + \mathrm{i} \cdot b_i)$ with $b_i \in [-1, 1]$.

Let us show that z belongs to the range if and only if the given instance of the partition problem has a solution. Indeed, for each i, the product $z_i \cdot t_i$ is equal to

$$s_i \cdot (1 + \mathrm{i} \cdot a_i) \cdot (1 + \mathrm{i} \cdot b_i) = s_i \cdot ((1 - a_i \cdot b_i) + \mathrm{i} \cdot (a_i + b_i)).$$

Since $|a_i| \le 1$ and $|b_i| \le 1$, we have $|a_i \cdot b_i| \le 1$, and therefore, $1 - a_i \cdot b_i \ge 0$. Thus, the real part of the sum $\sum_{i=1}^{n} z_i \cdot t_i$ is equal to the sum of n non-negative numbers $s_i \cdot (1 - a_i \cdot b_i)$. The only possibility for this sum to be equal to 0 if when all n non-negative terms are equal to 0, i.e., when $a_i \cdot b_i = 1$.

Since $|a_i| \le 1$ and $|b_i| \le 1$, the absolute value of the product $|a_i \cdot b_i|$ cannot exceed 1, and the only possibility for this product to be equal to 1 is when both absolute values are equal to 1, i.e., when $a_i = \pm 1$ and $b_i = \pm 1$. Since $a_i \cdot b_i = 1$, the signs must coincide, i.e., we must have $a_i = b_i = \varepsilon_i \in \{-1, 1\}$. For these values a_i and b_i, the imaginary part of $z_i \cdot t_i$ is equal to $2 \cdot \varepsilon_i \cdot s_i$, so the fact that the imaginary part of the sum $\sum_{i=1}^{n} z_i \cdot t_i$ is equal to 0 is equivalent to $2 \cdot \sum_{i=1} \varepsilon_i \cdot s_i = 0$ – i.e., to the fact that the original instance of the partition problem has a solution.

The theorem is proven.

Proof of Theorem 4. This proof is similar to the proof of Theorem 3; the only difference is that here, for $z_i = \sqrt{s_i} \cdot (1 + \mathrm{i} \cdot a_i)$, we have $z_i^2 = s_i \cdot ((1 - a_i^2) + \mathrm{i} \cdot (2a_i))$. Since $|a_i| \le 1$, we have $1 - a_i^2 \ge 0$, and the only possibility for the expression $\frac{1}{n} \cdot \sum_{i=1}^{n} z_i^2$ to have a zero real part is to have $1 - a_i^2 = 0$ for all i, i.e., to have $a_i = \pm 1$ for every i. In other words, we have $a_i = \varepsilon_i \in \{-1, 1\}$. For these a_i, the imaginary part of the expression $\frac{1}{n} \cdot \sum_{i=1}^{n} z_i^2$ is equal to $\frac{2}{n} \cdot \sum_{i=1}^{n} s_i \cdot \varepsilon_i$. Thus, the fact that the imaginary part is equal to 0 is equivalent to $\sum_{i=1}^{n} \varepsilon_i \cdot s_i = 0$, i.e., to the existence of the solution to the original instance of the partition problem.

Proof of Theorem 5. We will prove that computing the largest possible real part $\overline{x}$ of the product is NP-hard.

In this proof, we will use the same reduction as in Theorem 1. In this proof, we showed that when $z_i \in \mathbf{z}_i$, the product $z = z_1 \cdot \ldots \cdot z_n$ has a magnitude which cannot exceed $z \stackrel{\text{def}}{=} \prod_{i=1}^{n} \sqrt{1 + t_i^2}$. Since the real part of a complex number cannot exceed its magnitude, the largest possible value $\overline{x}$ of the real part cannot exceed z. The only possibility for $\overline{x}$ to be equal to z is when there is a point with real value z; since the magnitude cannot exceed z, the imaginary part of this point must be 0. Thus, the only way for $\overline{x}$ to be equal to z is to have z itself represented as a product of values $z_i \in \mathbf{z}_i$, and we already know that checking this condition is NP-hard. Thus, computing $\overline{x}$ is also NP-hard.

Proof of Theorem 6. Similarly to the proof of Theorem 3, take $\mathbf{z}_i = \mathbf{t}_i = \sqrt{s_i} \cdot \left(1, \frac{\sqrt{2}}{2}\right)$ and $z = 0$. One can easily check that for complex numbers $z_i \in \mathbf{z}_i$, the phase takes the values from -45° to 45°. The phase is equal to 45° only at a point $\sqrt{s_i} \cdot (0.5 + \mathrm{i} \cdot 0.5)$, and the phase is equal to -45° only at a point $\sqrt{s_i} \cdot (0.5 - \mathrm{i} \cdot 0.5)$.

When we multiply complex numbers, their phases add up. Thus, for the product $z_i \cdot t_i$, the angle is always between -90° and 90°, i.e., the real part of the product is always non-negative. So, the real part of the sum $\sum_{i=1}^{n} z_i \cdot t_i$ is also always non-negative. The only possibility for this real part to be 0 is when the real parts of all the terms in the sum are equal to 0, i.e., when for each i, the phase of the product $z_i \cdot t_i$ is equal to either 90° or to -90°. This, in turn, is possible only if either both z_i and t_i have phases 45° or both z_i and t_i have phases -45°.

- In the first case, we have $z_i = t_i = \sqrt{s_i} \cdot (0.5 + \mathrm{i} \cdot 0.5)$, hence $z_i \cdot t_i = 0.5 \cdot s_i \cdot \mathrm{i}$.
- In the second case, we have $z_i = t_i = \sqrt{s_i} \cdot (0.5 - \mathrm{i} \cdot 0.5)$, hence $z_i \cdot t_i = -0.5 \cdot s_i \cdot \mathrm{i}$.

In both cases, we have $z_i \cdot t_i = 0.5 \cdot \varepsilon_i \cdot s_i \cdot \mathrm{i}$ for some $\varepsilon_i \in \{-1, 1\}$.

Thus, the imaginary part of the sum $\sum_{i=1}^{n} z_i \cdot t_i$ is equal to $0.5 \cdot \sum_{i=1}^{n} \varepsilon_i \cdot s_i$. This imaginary part is equal to 0 if and only if $\sum_{i=1}^{n} \varepsilon_i \cdot s_i = 0$, i.e., if and only if the original instance of the particion problem has a solution.

This completes the proof of the theorem.

Proof of Theorem 7 is similar.

5. ACKNOWLEDGMENTS

This work was supported in part by NSF grants EAR-0225670 and DMS-0532645 and by Texas Department of Transportation grant No. 0-5453.

The authors are thankful to the participants of 11th GAMM–IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN'2004 (Fukuoka, Japan, October 4–8, 2004) for inspiring discussions that led to this work.

6. REFERENCES

[1] M. R. Garey and D. S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, California, 1979.

[2] E. Hansen, "Sharpness in interval computations", *Reliable Computing*, 1997, Vol. 3, pp. 7–29.

[3] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer, London, 2001.

[4] L. V. Kolev, *Interval Methods for Circuit Analysis*, World Scientific, Singapore, 1993.

[5] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1997.

[6] M. S. Petkovic and L. D. Petkovic, *Complex Interval Arithmetic and Its Applications*, J. Wiley, New York, 1998.