# A robust algorithm for noisy triangulation

– A most preliminary, partial report for Luiz to read and comment, modify etc –

Warwick Tucker
warwick@math.cornell.edu

January 24, 2002

**Abstract**

We present an algorithm developed to localize an acoustic beacon using several hydrophones. The problem can be generalized to any finite dimension $n$. The direct approach triangulates the position of the beacon by localizing the intersection of several $n$-spheres. This can be interpreted as a root-finding problem, and can thus be solved by Newton's method. Our approach reduces the problem to a system of linear equations, which can be solved by standard linear algebra techniques. To account for noisy inputs from the hydrophones, we develop an algorithm that utilizes interval arithmetic to achieve a rigorous and tight enclosure of the beacon's exact location.

## 1 Introduction

Consider a lake in which we have submerged $k$ hydrophones, located at the points $h_1, h_2, \ldots, h_k$, expressed in some external three-dimensional coordinate system. Now suppose that there is a beacon hidden in the lake at the point $b$. The beacon occasionally emits an acoustic signal (a ping) which travels through the water and triggers the hydrophones at the times $t_1, t_2, \ldots, t_k$ (relative to some external clock). Given the information $h_i, t_i$, $i = 1, \ldots, k$, how do we determine the location $b$ of the beacon? Indeed, under which conditions does a solution to our problem even exist?

We will prove that triangulating the location of a beacon in $\mathbb{R}^n$ is equivalent to solving a $(n+1)$-dimensional linear system of equations, whose elements are provided by data from $n+2$ hydrophones. In real-life applications, however, things are not so nice: the A/D converters in the hydrophones are not perfect, and thus introduce errors in the registered activation times. Also, the exact location of the hydrophones may not be known. The temperature of water is not uniform, and thus the signal travels with non-uniform speed. In this situation, we clearly can not find the exact location of the beacon. Given the tolerances of the hyrophones' locations and the estimated maximal errors in the recorded times, what we can strive for is to find a *region* guaranteed to contain the beacon.

The traditional way to approach this new situation is to perform a classical error analysis, examining every step of the computations outlined in the previous section. In most cases, this is extremely time-consuming, and often prone to error (no pun intended). We propose a radically different approach: instead of carrying out the error analysis by hand, we perform all computations in such a way so that all errors are automatically accounted for. Such computations can be realized by using interval arithmetic with directed rounding, as described in Section 4.

# 2 The two-dimensional case

For simplicity, let us discuss the two-dimensional case. Here the lake is just a bounded region in the plane, which we will also assume is convex. Such a scenario is illustrated in Figure 1.

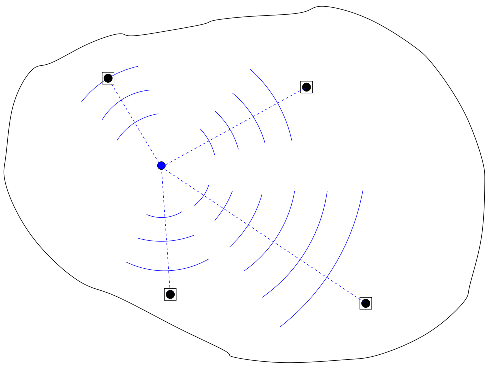

Figure 1: A lake with four hydrophones and one hidden beacon.

Assuming that the surrounding water is uniform and that the hydrophones work perfectly (i.e., there are no delays in the hydrophones' sensors), we should be able to pin-point the location of the beacon if we have at least three hydrophones at our disposal. There is one special case in which two hydrophones will suffice, namely when the beacon is located exactly on the straight line spanned by two hydrophones, except on the segment joining the two hydrophones. In all other cases, we need a third hydrophone (which must not be located on the abovementioned line) to determine the location of the beacon.

Considering the problem from the listener's point of view, it is clear that the hydrophone associated with the first time of detection, must also be the one located closest to the beacon. Let us call this pair $(h_{\hat{\imath}}, t_{\hat{\imath}})$. Since we are assuming that the speed of sound in water $v_w$ is constant, the time delay $\Delta t_{i,j} = t_i - t_j$ is directly proportional to the difference in distance from the two hydrophones $h_i$ and $h_j$ to the beacon, see Figure 2(a). Seeing that the distance between the beacon and the closest hydrophone is given by $r_{\hat{\imath}} = \|b - h_{\hat{\imath}}\|$, we know that

$$r_i = \|b - h_i\| = r_{\hat{\imath}} + \Delta t_{i,\hat{\imath}} v_w \qquad i = 1, \ldots, k. \tag{1}$$

Thus for $i = 1, \ldots, k$, we can center a circle $\gamma_i$ with radius $r_i$ around the hydrophone $h_i$. The crucial observation is that, according to (1), these radii only depend on one (unknown) parameter, namely $r_{\hat{\imath}}$. As this parameter is varied, all other circles are varied too. It is now a matter of finding a value $r^*_{\hat{\imath}}$ such that all circles intersect at one single point. This point will be the location of the beacon, i.e.,

$$b = \bigcap_{i=1}^{k} \gamma_i(r^*_{\hat{\imath}}).$$

The situation is illustrated in Figure 2(b).

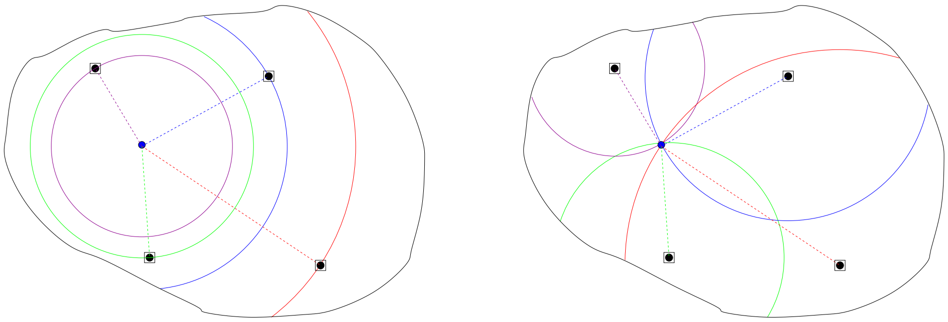

Figure 2: Seen from (a) the beacon; (b) the hydrophones.

How do we find the parameter $r^*_{\hat{\imath}}$ that gives a unique intersection? Finding the intersection of circles is equivalent to solving a non-linear system of equations, which in general requires Newton's method. Thus, in the 2-dimensional case, we would need to solve a 3-dimensional non-linear system of equations.

# 3 Locating a beacon in $\mathbb{R}^n$

Instead of continuing with our low-dimensional example, let us consider an arbitrary dimension, $n$. Now we are interested in finding the unique intersection of $n+1$ $n$-spheres, each centered around a single hydrophone, and all of whose radii depend only on a scalar parameter. Thus, in the $n$-dimensional case, we need to solve a $(n+1)$-dimensional non-linear system of equations.

Rather than following this approach, we will reduce our computations to a linear system of equations. The price we have to pay for this reduction is the addition of one more hydrophone. As we shall see, the resulting linear system is still $(n+1)$-dimensional. The additional hydrophone is clearly acceptable, since solving a *linear* system is in general much easier than solving a *non-linear* system of equations.

By (1), we have

$$\|b-h_i\| = r_{\hat\imath} + \Delta t_{i,\hat\imath} v_w = r_{\hat\imath} + \Delta r_{i,\hat\imath}, \tag{2}$$
$$\|b-h_j\| = r_{\hat\imath} + \Delta t_{j,\hat\imath} v_w = r_{\hat\imath} + \Delta r_{j,\hat\imath}, \tag{3}$$

where we use the short-hand $\Delta r_{i,j} = \Delta t_{i,j} v_w$. Note that the square of (2) is

$$\|b-h_i\|^2 = \|b\|^2 + \|h_i\|^2 - 2\langle b, h_i\rangle = \big(r_{\hat\imath}\big)^2 + \big(\Delta r_{i,\hat\imath}\big)^2 + 2r_{\hat\imath}\Delta r_{i,\hat\imath}.$$

Therefore, subtracting the squares of equations (2) and (3) gives

$$\|h_i\|^2 - \|h_j\|^2 - 2\langle b, h_i - h_j\rangle = \big(\Delta r_{i,\hat\imath} + \Delta r_{j,\hat\imath}\big)\Delta r_{i,j} + 2r_{\hat\imath}\Delta r_{i,j}, \tag{4}$$

where we have used the relation $\Delta r_{i,\hat\imath} - \Delta r_{j,\hat\imath} = \Delta r_{i,j}$. Note that the only unknowns in (4) are $b$ (an $n$-dimensional vector) and $r_{\hat\imath}$ (a scalar). Assuming that we have exactly $n+2$ hydrophones, we have enough data to reconstruct the location of the beacon. In order to avoid redundancies, we only want to consider certain index pairs $(i,j) = (\nu, \mu)$ in (4). The following choice is easy to remember:

$$(\nu(k), \mu(k)) = (k, k+1), \quad k = 1, \dots, n+1.$$

Next, for $i = 1, \dots, n+1$ and $j = 1, \dots, n$, we define the following constants:

$$a_{i,j} = -2\big((h_{\nu(i)})_j - (h_{\mu(i)})_j\big), \tag{5}$$
$$c_i = \big(\Delta r_{\nu(i),\hat\imath} + \Delta r_{\mu(i),\hat\imath}\big)\Delta r_{\nu(i),\mu(i)} - \big(\|h_{\nu(i)}\|^2 - \|h_{\mu(i)}\|^2\big), \tag{6}$$
$$d_i = 2\Delta r_{\nu(i),\mu(i)}. \tag{7}$$

We can now rewrite (4) as the following system of equations:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ a_{2,1} & \cdots & a_{2,n} \\ \vdots & \ddots & \vdots \\ a_{n+1,1} & \cdots & a_{n+1,n} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n+1} \end{pmatrix} + r_{\hat\imath} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n+1} \end{pmatrix}, \tag{8}$$

Finally, if we set $a_{i,n+1} = -d_i$ for $i = 1, \dots, n+1$, and $b_{n+1} = r_{\hat\imath}$, we get a $(n+1)$-dimensional linear system:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n+1} \\ a_{2,1} & \cdots & a_{2,n+1} \\ \vdots & \ddots & \vdots \\ a_{n+1,1} & \cdots & a_{n+1,n+1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n+1} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n+1} \end{pmatrix}, \tag{9}$$

where the elements of the matrix $A$ and the vector $c$ are known. The location of the beacon $(b_1, \dots, b_n)$ and the distance $b_{n+1}$ to the closest hydrophone $h_{\hat\imath}$ can be computed by standard methods, provided $A$ is invertible.

# 4 Interval arithmetic

In this section, we will briefly describe the fundamentals of interval arithmetic. For a concise reference on this topic, see [FS97], [Mo66] or [Mo79].

Let $\mathbb{IR}$ denote the set of all closed intervals of the real line. For any element $[a] \in \mathbb{IR}$, we adopt the notation $[a] = [\underline{a}, \bar{a}]$, and we allow for degenerate intervals $[a]$ with $\underline{a} = \bar{a}$. We define binary arithmetic operations on elements of $\mathbb{IR}$ in a set theoretic manner:

**Definition 4.1** *If $\odot$ is one of the operators $+, -, \times, \div$, we define arithmetic operations on elements of $\mathbb{IR}$ by*

$$[a] \odot [b] = \{a \odot b \colon a \in [a], b \in [b]\},$$

*except that $[a] \div [b]$ is undefined if $0 \in [b]$*[1].

Working exclusively with closed intervals, we can describe the resulting interval in terms of the endpoints of the operands:

$$\begin{aligned}
[a] + [b] &= [\underline{a} + \underline{b}, \bar{a} + \bar{b}] \\
[a] - [b] &= [\underline{a} - \bar{b}, \bar{a} - \underline{b}] \\
[a] \times [b] &= [\min\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}] \\
[a] \div [b] &= [a] \times [1/\bar{b}, 1/\underline{b}], \quad \text{if } 0 \notin [b].
\end{aligned}$$

It follows from the definition that addition and multiplication are both associative and commutative. Also, it is clear that the elements $[0, 0]$ and $[1, 1]$ are the unique neutral elements with respect to addition and multiplication, respectively.

Surprisingly, the distributive law does *not* always hold. As an example, we have

$$[-1, 1]([-1, 0] + [3, 4]) = [-1, 1][2, 4] = [-4, 4],$$

whereas

$$[-1, 1][-1, 0] + [-1, 1][3, 4] = [-1, 1] + [-4, 4] = [-5, 5].$$

This unusual property is important to keep in mind when representing functions as part of a program. Interval arithmetic satisfies a weaker rule than the distributive law, which we shall refer to as *sub-distributivity*:

$$[a]([b] + [c]) \subseteq [a][b] + [a][c].$$

This is a set theoretical property that illustrates one of the fundamental differences between real- and interval arithmetic.

Another key feature of interval arithmetic is that of *inclusion monotonicity*:

[1]We could allow for division by zero by admitting intervals on the form $[-\infty, x]$, $[x, +\infty]$, and $[-\infty, +\infty]$. With this *extended* interval arithmetic, we can compute e.g. $[1, 2] \div [0, 5] = [1/5, +\infty]$.

**Theorem 4.2** *If* $[a] \subseteq [a']$, $[b] \subseteq [b']$, *and* $\odot \in \{+, -, \times, \div\}$, *then*

$$[a] \odot [b] \subseteq [a'] \odot [b'],$$

*where we demand that* $0 \notin [b']$ *for division.*

This is the single most important property of interval arithmetic: it allows us to prove open conditions in a robust way.

We can turn $\mathbb{IR}$ into a metric space by equipping it with the Hausdorff distance:

$$d([a], [b]) = \max\{|\underline{a} - \underline{b}|, |\bar{a} - \bar{b}|\}.$$

For dealing with higher dimensional problems, we define the arithmetic operations to be carried out component-wise. We then talk about an *interval vector* or, more simply, a *box*. The metric on the space $\mathbb{IR}^n$ is defined by

$$d([a], [b]) = \max_{1 \le i \le n} \{d([a_i], [b_i])\}.$$

When implementing interval arithmetic on a computer, we no longer work over the space $\mathbb{R}$, but rather $\mathbb{F}$ - the floating points of the computer. This is a finite set, and thus so is $\mathbb{IF}$ – the set of all intervals with floating point endpoints:

$$\mathbb{IF} = \{[\underline{a}, \bar{a}] \colon \underline{a} \le \bar{a}; \quad \underline{a}, \bar{a} \in \mathbb{F}\}.$$

When performing arithmetic on intervals in $\mathbb{F}$ we must round the resulting interval outwards to guarantee inclusion of the true result. This is because $\mathbb{F}$ is not arithmetically closed. For $[a], [b] \in \mathbb{IF}$, we define

$$\begin{aligned}
[a] + [b] &= [\downarrow \underline{a} + \underline{b} \downarrow, \uparrow \bar{a} + \bar{b} \uparrow] \\
[a] - [b] &= [\downarrow \underline{a} - \bar{b} \downarrow, \uparrow \bar{a} - \underline{b} \uparrow] \\
[a] \times [b] &= [\min\{\downarrow \underline{a}\underline{b} \downarrow, \downarrow \underline{a}\bar{b} \downarrow, \downarrow \bar{a}\underline{b} \downarrow, \downarrow \bar{a}\bar{b} \downarrow\}, \\
&\qquad \max\{\uparrow \underline{a}\underline{b} \uparrow, \uparrow \underline{a}\bar{b} \uparrow, \uparrow \bar{a}\underline{b} \uparrow, \uparrow \bar{a}\bar{b} \uparrow\}] \\
[a] \div [b] &= [a] \times [\downarrow 1/\bar{b} \downarrow, \uparrow 1/\underline{b} \uparrow], \quad \text{if } 0 \notin [b].
\end{aligned}$$

Recall that $\downarrow x \downarrow$ is the largest floating point in $\mathbb{F}$ that is less than or equal to $x$ (called $x$ rounded down), and $\uparrow x \uparrow$ is the smallest floating point in $\mathbb{F}$ that is greater than or equal to $x$ (called $x$ rounded up). This type of arithmetic is called interval arithmetic with *directed rounding*.

# 5 Inverting interval matrices

In this section, we will extend matrix operations to include interval-valued matrices and vectors. An interval-valued matrix is simply a matrix whose components are intervals. Given two interval matrices $[A]$ and $[B]$ of equal size, we say that $[A] \subseteq [B]$ iff $[A_{i,j}] \subseteq [B_{i,j}]$ for all components. With this notation, it makes sense to write $A \in [B]$ for a real-valued matrix $A$ with $a_{i,j} \subseteq [B_{i,j}]$.

Addition, subtraction, and multiplication by a scalar can be defined componentwise, and will produce the exact pointwise ranges:

$$\begin{aligned} [A] \pm [B] &= \{A \pm B \colon A \in [A]; B \in [B]\}, \\ c[A] &= \{cA \colon A \in [A]\}. \end{aligned}$$

We define interval matrix multiplication analogously to the real-valued situation:

**Definition 5.1** *Consider two interval matrices $[A]$ and $[B]$ having sizes $n \times m$ and $m \times k$, respectively. Their product $[C] = [A][B]$ is then defined by*

$$[c_{i,j}] = \sum_{l=1}^{m} [a_{i,l}][b_{l,j}] \qquad (i = 1, \ldots, n; j = 1, \ldots, k).$$

In particular, for an interval vector $[x] \in \mathbb{IR}^m$, the image $[y] = [A][x]$ is defined by $[y_i] = \sum_{k=1}^{m} [a_{i,k}][x_k]$, $i = 1, \ldots, n$.

Note that, as opposed to when computing with single intervals (see Definition 4.1), matrix multiplication will generally produce strict inclusions of the pointwise range:

$$\begin{aligned} [A][B] &\supseteq \{AB \colon A \in [A]; B \in [B]\}, \\ [A][x] &\supseteq \{Ax \colon A \in [A]; x \in [x]\}. \end{aligned}$$

Even multiplication by an interval scalar may produce an over-estimation:

$$[c][A] \supseteq \{cA \colon A \in [A]; c \in [c]\},$$

as the following example illustrates.

Example 1

Consider the interval matrix

$$[A] = \begin{pmatrix} [1,2] & 3 \\ 4 & 5 \end{pmatrix}.$$

If we take the interval scalar $[c] = [-1, 1]$, then we have

$$[c][A] = \begin{pmatrix} [-2,2] & [-3,3] \\ [-4,4] & [-5,5] \end{pmatrix} \ni \begin{pmatrix} -2 & 3 \\ 4 & 5 \end{pmatrix} \notin \{cA \colon A \in [A]; c \in [c]\}.$$

$\triangle$

In matrix computations, the concept of division is treated through multiplication by matrix inverses, when these are well-defined. Of course, we are interested in generalizing the concept of inversion to interval matrices. Thus, given an interval matrix $[A]$, how do we compute $[A]^{-1}$? First, we must define what we *mean* by $[A]^{-1}$. In the real case, $A^{-1}$ is the matrix that satisfies $I = AA^{-1} = A^{-1}A$. As is well-known, the existence of $A^{-1}$ requires that $A$ is a square matrix with non-zero determinant. Motivated by this, we define its inverse to be the set of all inverses of real matrices $B$ in $[A]$:

**Definition 5.2** *Given an interval matrix $[A]$, we define the interval matrix inverse*

$$[A]^{-1} = \{B^{-1}\colon B \in [A]\}, \tag{10}$$

*whenever this is well-defined.*

It follows that, for $[A]$ to be invertible, it must not contain a single non-invertible real matrix. If this condition is satisfied, we say that $[A]$ is *non-singular* or simply *invertible*. Note that, in general, we have $A[A]^{-1} \neq [A]^{-1}A$. All we know is that both interval matrices contain the identity matrix.

Unfortunately, utilizing exact formulas for computing matrix inverses may result in large overestimations when the computations are performed in interval arithmetic.

EXAMPLE 2

Consider the interval matrix

$$[A] = \begin{pmatrix} 1 & -2 \\ 1 & [1, \frac{3}{2}] \end{pmatrix},$$

whose inverse we would like to compute. Using the formula for $2 \times 2$ matrices:

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \Rightarrow \quad M^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

we arrive at

$$[A]^{-1} = \frac{1}{[1, \frac{3}{2}] + 2}\begin{pmatrix} [1, \frac{3}{2}] & 2 \\ -1 & 1 \end{pmatrix}.$$

In particular, concentrating on the first component, we have

$$([A]^{-1})_{1,1} = \frac{[1, \frac{3}{2}]}{[1, \frac{3}{2}] + 2} = [1, \tfrac{3}{2}] \div [3, \tfrac{7}{2}] = [\tfrac{2}{7}, \tfrac{1}{2}],$$

which overestimates the true range. Indeed, according to (5.2), the exact range is

$$([A]^{-1})_{1,1} = \left\{\frac{x}{x+2}\colon x \in [1, \tfrac{3}{2}]\right\} = [\tfrac{3}{7}, \tfrac{1}{2}].$$

$\triangle$

This defect should come as no surprise: indeed, this overestimation *must* occur when computing the range of a function where a variable occurs more than once. Thus what remains is to device a method for *accurately* computing $[A]^{-1}$. Before describing an algorithm for inverting an interval matrix, we will define the *maximal row sum norm* of an $n \times n$ interval matrix:

$$\|[A]\| = \max_{1 \le i \le n} \sum_{j=1}^{n} |[a_{i,j}]|,$$

where we use the notation $|[a_{i,j}]| = \max\{|\underline{a}_{i,j}|, |\bar{a}_{i,j}|\}$. Note that, for all $A \in [A]$, we have $\|A\| \le \|[A]\|$.

**Theorem 5.3 (Interval matrix inversion)** *Let $[A]$ be a non-singular $n \times n$ interval matrix, and set $[E] = I - [A]Y$, where $Y$ is close to the inverse of some real matrix $X \in [A]$. If $\|[E]\| \leq r < 1$, then we can enclose the inverse interval matrix:*

$$[A]^{-1} \subseteq Y([S_k] + [P_k]).$$

*Here $[P_k]$ is the interval matrix with all entries equal to $[-r^{k+1}, r^{k+1}]/(1-r)$, and*

$$[S_k] = I + [E] + [E]^2 + \cdots + [E]^k.$$

The proof, which is due to Hansen (see [Ha65]), is motivated by the fact that if we set $E = I - AY$, then

$$Y(I-E)^{-1} = Y(I-(I-AY))^{-1} = Y(AY)^{-1} = YY^{-1}A^{-1} = A^{-1}.$$

This requires that both $A$ and $Y$ are non-singular, the former which we assume, and the latter which we attempt to ensure by taking $Y$ close to $A^{-1}$.

# 6 Applications

Let us return to our original problem: the triangulation of an acoustic beacon using hydrophones. In the real-world case, instead of solving a perfect linear system of equations $Ab = c$, we have an interval system: $[A]b \in [c]$. In other words, we want to find the set of all $b$s such that $[A]b \in [c]$. By the definition of the inverse of an interval matrix, we have the inclusion.

$$\{b = A^{-1}c \colon A \in [A]; c \in [c]\} \subseteq [A]^{-1}[c].$$

This means that given $[A]$ and $[c]$, the beacon is guaranteed to be contained in the box $[X][c]$, where $[X]$ is an enclosure of $[A]^{-1}$.

# References


[FS97] de Figueiredo, L. H., Stolfi, J., Métodos numéricos auto-validados e aplicações, Braz. Math. Colloq. **21**, IMPA, Rio de Janeiro, 1997.

[Ha65] Hansen, E., *Interval Arithmetic in Matrix Computations*, J. SIAM Numer. Anal., Ser. B **2:2** (1965), 308-320.

[Mo65] Moore, R. E., *The Automatic Analysis and Control of Error in Digital Computations Based on the Use of Interval Numbers*, in Error in Digital Computing, Vol 1, John Wiley & Sons Inc, New York, 1965.

[Mo66] Moore, R. E., Interval Analysis. Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[Mo79] Moore, R. E., Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia, 1979.