INTERVAL METHODS IN MATHEMATICAL PROGRAMMING

R. Baker Kearfott

Department of Mathematics

University of Southwestern Louisiana

Lafayette, LA

email rbk@ucs.usl.edu

and Vladik Kreinovich

Department of Computer Science

University of Texas at El Paso

El Paso, TX

email vladik@cs.ep.utexas.edu

Problem

- **Problem:**

  Traditional methods of numerical mathematics result in approximate solutions of numerical problems, often with *no guaranteed estimates* of the solution's inaccuracy.

- In many real applications, knowing such estimates is really important.

- **Example.**

  If according to the (approximately computed) trajectory of a spaceship it will land in the desired area, then whether it will actually land or crush depends on the inaccuracy of this numerical prediction.

- **Methods:**

  In view of that, methods have been developed that provide us with reliable (=guaranteed) estimates. As a result of applying such methods, we get an approximate number A with a guarantee that the difference between the actual value and A does not exceed some given number Δ.

- **What does the name mean?** In other words, we guarantee that the actual value belongs to the interval $[A - \Delta, A + \Delta]$. Because of that, reliable methods are also called *methods of interval computations.*

Outline

In this short tutorial, we will describe the main ideas of this approach:

- naive interval computations
- centered form
- Hansen's approach (and its relation to nonstandard analysis)
- approximate interval methods:
  - methods based on sensitivity analysis
  - Monte-Carlo type methods based on Cauchy distribution (and why Cauchy)

Main problem

We know:

- an algorithm f that transforms n real numbers $x_1, ..., x_n$ into a real number $y = f(x_1, ..., x_n)$;

  This algorithm solves the desired problem in case we know the exact values of the data.

- the approximate values $\tilde{x}_1, ..., \tilde{x}_n$ of the parameters x_i, and the accuracies Δ_i of these approximate values

  (i.e., numbers such that $|\tilde{x}_i - x_i| \leq \Delta_i$*);*

- the only information we have about the actual values of x_i is that x_i belongs to an interval $[\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$.

We must find:

- the interval Y of possible values of $y = f(x_1, ..., x_n)$

  (or at least a good estimate F for that interval).

It is known: *Even for polynomial f, the problem of computing $f(X)$ exactly is NP-hard.*

Therefore, *we need fast methods of finding F such that $f(X) \subseteq F$.*

Naive interval computations: idea

- **Sum:**
  - $x_1 \in [a_1, b_1]$
  - $x_2 \in [a_2, b_2]$
  - $x_1 + x_2 \in ?$

    Answer: $x_1 + x_2 \in [a_1 + a_2, b_1 + b_2]$.
- **Difference:**
  - $x_1 - x_2 \in ?$

    Answer: $x_1 - x_2 \in [a_1 - b_2, b_1 - a_2]$.
- **Product:** $x_1 x_2 \in ?$

Answer: $x_1 x_2 \in [p^-, p^+]$, where

$p^- = \min(a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2)$ and

$p^+ = \max(a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2)$.

Naive interval computations: method

- **Example:** $f(x) = (x-2)(x+2)$, $x \in [1,2]$.
- How will the computer compute it?
  - $r_1 := x - 2$;
  - $r_2 := x + 2$;
  - $r_3 := r_1 * r_2$.
- **Main idea:** do the same operations, but with *intervals* instead of *numbers*:
  - $R_1 := [1,2] - [2,2] = [-1,0]$;
  - $R_2 := [1,2] + [2,2] = [3,4]$;
  - $R_3 := [-1,0] * [3,4] = [-4,0]$.
- **Actual range:** $f(X) = [-3,0]$.
- *Comment.* We always get a *guaranteed* estimate, but often a too large one.

Centered form

- **Main idea:**
  - represent $f(x)$ as a function of $\Delta x = x - \bar{x}$, where $\bar{x}$ is a center of X, and
  - apply naive interval computations to the resulting expression.
- **Example:**
  - $\bar{x} = 1.5$, $x = \Delta x + 1.5$, $\Delta x \in [-0.5, 0.5]$;
  - $(x-2)(x+2) = (\Delta x - 0.5)(\Delta x + 3.5) = \Delta x^2 + 3\Delta x - 1.75$;
  - For this expression, naive interval computations lead to $[-3.5, 0]$.
- **General property:** *asymptotically, when errors* $\to 0$, *it gives the correct error estimate* $(f(X) \sim F)$.

Hansen's approach

- **Main idea:**

  on each step, we represent the result of our computations as $a + a_0\Delta x_1 + ... + a_n\Delta x_n + A$, where

  - $\Delta x_i = x_i - \tilde{x}_i$, and
  - A is an interval that contains quadratic and other terms

- **Example:**

  - $x = 1.5 - \Delta x$;
  - $r_1 := x - 2 = -0.5 - \Delta x$;
  - $r_2 := x + 2 = 3.5 - \Delta x$;
  - $r_3 := r_1 * r_2 = (-0.5 - \Delta x)(3.5 - \Delta x) =$

    $= -1.75 - 3\Delta + [0, 0.25]$;
  - As a result, we get

    $F = -1.75 - 3[-0.5, 0.5] + [0, 0.25] = [-3.25, 0]$.

- **Computational complexity:**
  - **For naive interval computations:**

    ≤ 4 times more computations

  - **For Hansen's method:**

    n times more computations.

Approximate interval methods: main idea

- **Main assumption:**

  These methods are based on the usual physical assumption that we can neglect the terms that are quadratic in errors.

  Example: if we know the values x_i with the precision 2% (0.02), then the quadratic terms are proportional to 0.0004 (0.04%), and can be often safely neglected.

- **Resulting formula for Δy:** If we neglect quadratic and higher order terms in the expansion of

$$\Delta y = y - \tilde{y} = f(x_1, ..., x_n) - f(\tilde{x}_1, ...\tilde{x}_n) =$$

$$f(\tilde{x}_1 - \Delta x_1, ...\tilde{x}_n - \Delta x_n) - f(\tilde{x}_1, ..., \tilde{x}_n),$$

we get the expression

$$\Delta y = f_{,1}\Delta x_1 + ... + f_{,n}\Delta x_n,$$

where $f_{,i}$ denotes the partial derivative

$$\frac{\partial f}{\partial x_i}.$$

- **Resulting formula for Δ:**
  - **We know:** $|\Delta x_i| \leq \Delta_i$;
  - **We conclude:** $\Delta = |f_{,1}|\Delta_1 + .. + |f_{,n}|\Delta_n$.

Sensitivity analysis

- **Problem:**

  We are considering a complicated case, when an algorithm f is *not* simply an explicit expression, but a very complicated algorithm. So, it is impossible to differentiate f analytically.

- **Idea:**

  To use numerical estimates based on the same assumption (that the terms that are quadratic in errors are negligible).

- **Method:** Estimate $f_{,i}$ as

$$\frac{f(\tilde{x}_1, \ldots, \tilde{x}_{i-1}, \tilde{x}_i + h, \tilde{x}_{i+1}, \ldots, \tilde{x}_n) - \tilde{y}}{h}.$$

- **Computational complexity:**

  we apply f $n+1$ times:

  - first, to compute $\tilde{y}$,
  - and then n more times, to estimate the partial derivatives.
  - Then, we compute Δ.

- **Main drawback of sensitivity analysis:**

  - For many real-life problems (e.g., for the analysis of a geophysical data), the number of inputs n can be in thousands, and
  - each computation of f is (already) very time-consuming.
  - As a result, computing Δ takes *too much time*.

A Monte-Carlo-type method based on Cauchy distribution

- **Main steps:**
  - First, we *simulate* errors, i.e., use a computer random number generator to generate random numbers ξ_i that are distributed according to Cauchy distribution with a density

$$\rho(x) = \frac{\text{const}}{1 + (x/\Delta_i)^2}$$

with 0 average and (scale) parameter Δ_i.

  - Then, we compute $\Delta y^{(1)} = \tilde{y} - y^{(1)}$, where $y^{(1)} = f(\tilde{x}_1 - \xi_1, ..., \tilde{x}_n - \xi_n)$.

Comment. In the case when we can neglect terms that are quadratic in error, we can conclude that $\Delta y^{(1)}$ is a Cauchy-distributed random variable with 0 average and parameter

$$\Delta = \sum_i |f_{,i}| \Delta_i.$$

– So, to determine Δ, we repeat this procedure several times, obtaining N values

$$\Delta y^{(1)} = \tilde{y} - y^{(1)}, ..., \Delta y^{(N)} = \tilde{y} - y^{(N)},$$

and then apply standard statistical techniques (namely, Maximum Likelihood Method *MLM*) to estimate Δ.

– For Cauchy distribution, MLM turns into solving an equation

$$\sum_k \frac{1}{1 + (y^{(k)}/\Delta)^2} = \frac{N}{2}$$

(it can be solved by bisection).

- **Results:**

  For $N = 50$, we get Δ with a 20% accuracy in $\geq 99.9\%$ of cases.

  Comment. A 20% accuracy is quite sufficient if we take into consideration that this is a precision with which we know accuracy. There is little difference between a measuring device with a 2% accuracy and a device with a 2.1% accuracy.

- **Computational complexity:**

  – this method executes f $N + 1 = 51$ times.

  – So, for large n, this method is much *faster* than sensitivity analysis.

- **Main advantage over naive and centered interval methods:**
  - Naive and centered interval methods overestimate $f(X)$; often enormously;
  - This Monte-Carlo method gives an interval that (with a 99.9% guarantee) differs from the desired interval by $\leq 20\%$.

- **Both approximate methods may be easily parallelized.**
  - **Problem:** In all these methods, the *most time-consuming part* of the algorithm is applying a time-consuming algorithm f to different data.
  - **Idea:** So, a natural idea to save time is to make all these calls of f handled by separate processors.
  - **Result:** If we have several processors working in parallel, then *we may compute both the estimate $\tilde{y}$ and its accuracy in practically the same time that we would have spent on an estimate $\tilde{y}$ itself.*