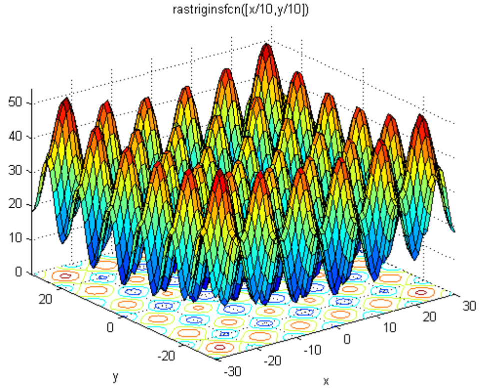
The Role of Interval Linear Algebra in Global Optimization

Milan Hladík

Department of Applied Mathematics,
Faculty of Mathematics and Physics,
Charles University,
Prague, Czech Republic
http://kam.mff.cuni.cz/~hladik/

Minisymposium on Trusted Numerical Computations
ETAMM2018
Kraków, June 19–20, 2018

Example



One of the Rastrigin functions.

Global Optimization Questions

Questions

- Can we find global minimum?
- Can we prove that the found solution is optimal?
- Can we prove uniqueness?
- Can we handle roundoff errors?

Bad news

- **No**, there is no algorithm solving global optimization problems using operations $+, \times, \sin$. [Zhu, 2005]

  (From Matiyasevich's theorem solving the 10th Hilbert problem.)

Good news

- **Yes** (under certain assumption) by using *Interval Computations*.

Interval Computations

Notation

An interval matrix

$$\boldsymbol{A} := [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m \times n} \mid \underline{A} \leq A \leq \overline{A}\}.$$

The center and radius matrices

$$A^c := \frac{1}{2}(\overline{A} + \underline{A}), \quad A^\Delta := \frac{1}{2}(\overline{A} - \underline{A}).$$

The set of all $m \times n$ interval matrices: $\mathbb{IR}^{m \times n}$.

Main Problem

Let $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{IR}^n$. Determine the image

$$f(\boldsymbol{x}) = \{f(x) : x \in \boldsymbol{x}\},$$

or at least its tight interval enclosure.

Interval Arithmetic

Interval Arithmetic (proper rounding used when implemented)

For arithmetical operations $(+, -, \cdot, /)$, their images are readily computed

$$\boldsymbol{a} + \boldsymbol{b} = [\underline{a} + \underline{b}, \overline{a} + \overline{b}],$$

$$\boldsymbol{a} - \boldsymbol{b} = [\underline{a} - \overline{b}, \overline{a} - \underline{b}],$$

$$\boldsymbol{a} \cdot \boldsymbol{b} = [\min(\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}), \max(\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b})],$$

$$\boldsymbol{a} / \boldsymbol{b} = [\min(\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b}), \max(\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b})], \quad 0 \notin \boldsymbol{b}.$$

Some basic functions $\boldsymbol{x}^2$, $\exp(\boldsymbol{x})$, $\sin(\boldsymbol{x})$, ..., too.

Can we evaluate every arithmetical expression on intervals?
Yes, but with overestimation in general due to dependencies.

Example (Evaluate $f(x) = x^2 - x$ on $\boldsymbol{x} = [-1, 2]$)

$$\boldsymbol{x}^2 - \boldsymbol{x} = [-1, 2]^2 - [-1, 2] = [-2, 5],$$

$$\boldsymbol{x}(\boldsymbol{x} - 1) = [-1, 2]([-1, 2] - 1) = [-4, 2],$$

$$(\boldsymbol{x} - \tfrac{1}{2})^2 - \tfrac{1}{4} = ([-1, 2] - \tfrac{1}{2})^2 - \tfrac{1}{4} = [-\tfrac{1}{4}, 2].$$

Global Optimization by Interval Techniques

Global optimization problem

Compute global (not just local!) optima to

$$\min \ f(x) \ \text{ subject to } \ g(x) \leq 0, \ h(x) = 0, \ x \in \boldsymbol{x}^0,$$

where $\boldsymbol{x}^0 \in \mathbb{IR}^n$ is an initial box.

Basic idea

- Split the initial box $\boldsymbol{x}^0$ into sub-boxes.
- If a sub-box does not contain an optimal solution, remove it. Otherwise split it into sub-boxes and repeat.

Interval Approach to Global Optimization

Branch & prune scheme

$\mathcal{L} := \{\boldsymbol{x}^0\}$, [set of boxes]
$c^* := \infty$, [upper bound on the minimal value]
while $\mathcal{L} \neq \emptyset$ **do**
  choose $\boldsymbol{x} \in \mathcal{L}$ and remove $\boldsymbol{x}$ from $\mathcal{L}$,
  contract $\boldsymbol{x}$,
  find a feasible point $x \in \boldsymbol{x}$ and update c^*,
  if $\max_i x_i^{\Delta} > \varepsilon$ **then**
    split $\boldsymbol{x}$ into sub-boxes and put them into $\mathcal{L}$,
  else
    give $\boldsymbol{x}$ to the output boxes,
  end if
end while

It is a rigorous method to enclose all global minima in a set of boxes.

Implementation Details

Which box to choose?

- the oldest one
- the one with the largest edge, i.e., for which $\max_i x_i^{\Delta}$ is maximal
- the one with minimal $\underline{f}(\boldsymbol{x})$.

How to divide the box?

- the widest edge
- the coordinate in which f varies possibly mostly (Walster, 1992; Ratz, 1992)

By Ratschek & Rokne (2009) there is no best strategy for splitting.

Contracting and Pruning $(\min f(x) \;\text{ s.t. }\; g(x) \le 0,\, h(x) = 0)$

Aim

Shrink $\boldsymbol{x}$ to a smaller box (or completely remove) such that no global minimum is removed.

Simple techniques

- if $0 \notin h_i(\boldsymbol{x})$ for some i, then remove $\boldsymbol{x}$
- if $0 < g_j(\boldsymbol{x})$ for some j, then remove $\boldsymbol{x}$
- if $0 < f'_{x_i}(\boldsymbol{x})$ for some i, then fix $\boldsymbol{x}_i := \underline{x}_i$
- if $0 > f'_{x_i}(\boldsymbol{x})$ for some i, then fix $\boldsymbol{x}_i := \overline{x}_i$

Optimality conditions

- employ the Fritz–John (or the Karush–Kuhn–Tucker) conditions

$$\begin{aligned} u_0 \nabla f(x) + u^T \nabla h(x) + v^T \nabla g(x) &= 0, \\ h(x) = 0, \quad v_\ell g_\ell(x) = 0 \;\forall \ell, \quad \|(u_0, u, v)\| &= 1. \end{aligned}$$

- solve by the Interval Newton method

Inside the feasible region

Suppose there are no equality constraints and $g_j(\boldsymbol{x}) < 0\ \forall j$.

- (monotonicity test) if $0 \notin f'_{x_i}(\boldsymbol{x})$ for some i, then remove $\boldsymbol{x}$
- apply the Interval Newton method to the additional constraint $\nabla f(x) = 0$
- (nonconvexity test) if the interval Hessian $\nabla^2 f(\boldsymbol{x})$ contains no positive semidefinite matrix, then remove $\boldsymbol{x}$

Contracting and Pruning

Constraint propagation

Iteratively reduce domains for variables such that no feasible solution is removed by handling the relations and the domains.

Example

Consider the constraint

$$x + yz = 7, \quad x \in [0, 3],\ y \in [3, 5],\ z \in [2, 4]$$

- express x

  $$x = 7 - yz \in 7 - [3, 5][2, 4] = [-13, 1]$$

  thus, the domain for x is $[0, 3] \cap [-13, 1] = [0, 1]$

- express y

  $$y = (7 - x)/z \in (7 - [0, 1])/[2, 4] = [1.5,\ 3.5]$$

  thus, the domain for y is $[3, 5] \cap [1.5,\ 3.5] = [3,\ 3.5]$

Lower and Upper Bounds

For each Branch & Bound algorithm are essential:

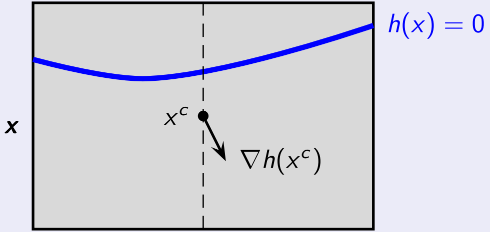
- tight upper bounds
- tight lower bounds

Upper Bounds – Feasibility Test

Aim

Find a feasible point x^*, and update $c^* := \min(c^*, f(x^*))$.

- if no equality constraints, take e.g. $x^* := x^c$
- if k equality constraints, fix $n - k$ variables $x_i := x_i^c$ and solve system of equations by the interval Newton method
- if $k = 1$, fix the variables corresponding to the smallest absolute values in $\nabla h(x^c)$



Upper Bounds – Feasibility Test

Aim

Find a feasible point x^*, and update $c^* := \min(c^*, f(x^*))$.

- if no equality constraints, take e.g. $x^* := x^c$
- if k equality constraints, fix $n - k$ variables $x_i := x_i^c$ and solve system of equations by the interval Newton method
- if $k = 1$, fix the variables corresponding to the smallest absolute values in $\nabla h(x^c)$
- in general, if $k > 1$, transform the matrix $\nabla h(x^c)$ to a row echelon form by using a complete pivoting, and fix components corresponding to the right most columns
- we can include $f(x) \leq c^*$ to the constraints

Lower Bounds

Aim

Given a box $\boldsymbol{x} \in \mathbb{IR}^n$, determine a lower bound to $\underline{f}(\boldsymbol{x})$.

Why?

- if $\underline{f}(\boldsymbol{x}) > c^*$, we can remove $\boldsymbol{x}$
- minimum over all boxes gives a lower bound on the optimal value

Methods

- interval arithmetic
- mean value form
- Lipschitz constant approach
- αBB algorithm
- ...

Intermezzo – Eigenvalues of Symmetric Interval Matrices

A symmetric interval matrix

$$\boldsymbol{A}^S := \{A \in \boldsymbol{A} \colon A = A^T\}.$$

Without loss of generality assume that $\underline{A} = \underline{A}^T$, $\overline{A} = \overline{A}^T$, and $\boldsymbol{A}^S \neq \emptyset$.

Eigenvalues of a symmetric interval matrix

Eigenvalues of a symmetric $A \in \mathbb{R}^{n \times n}$: $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$.

Eigenvalue sets of $\boldsymbol{A}^S$ are compact intervals

$$\boldsymbol{\lambda}_i(\boldsymbol{A}^S) := \left\{\lambda_i(A) \colon A \in \boldsymbol{A}^S\right\}, \quad i = 1, \ldots, n.$$

Theorem

Checking whether $0 \in \boldsymbol{\lambda}_i(\boldsymbol{A}^S)$ *for some* $i = 1, \ldots, n$ *is NP-hard.*

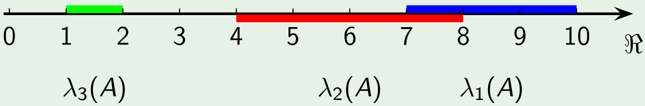
Eigenvalues – An Example

Example

Let

$$A \in \boldsymbol{A} = \begin{pmatrix} [1,2] & 0 & 0 \\ 0 & [7,8] & 0 \\ 0 & 0 & [4,10] \end{pmatrix}$$

What are the eigenvalue sets?
We have $\boldsymbol{\lambda}_1(\boldsymbol{A}^S) = [7,10]$, $\boldsymbol{\lambda}_2(\boldsymbol{A}^S) = [4,8]$ and $\boldsymbol{\lambda}_3(\boldsymbol{A}^S) = [1,2]$.



Eigenvalue sets are compact intervals. They may intersect or equal.

Eigenvalues – Some Exact Bounds

Theorem (Hertz, 1992)

We have

$$\overline{\lambda}_1(\boldsymbol{A}^S) = \max_{z \in \{\pm 1\}^n} \lambda_1(A^c + \operatorname{diag}(z) A^\Delta \operatorname{diag}(z)),$$
$$\underline{\lambda}_n(\boldsymbol{A}^S) = \min_{z \in \{\pm 1\}^n} \lambda_n(A^c - \operatorname{diag}(z) A^\Delta \operatorname{diag}(z)).$$

Theorem

$\underline{\lambda}_1(\boldsymbol{A}^S)$ *and* $\overline{\lambda}_n(\boldsymbol{A}^S)$ *are polynomially computable by semidefinite programming with arbitrary precision.*

Proof.

We have

$$\overline{\lambda}_n(\boldsymbol{A}^S) = \max\ \alpha \ \text{ subject to }\ A - \alpha I_n \text{ is positive semidefinite},\ A \in \boldsymbol{A}^S.$$

$\square$

Eigenvalues – Easy Cases and Enclosures

Theorem

1. *If A^c is essentially non-negative, i.e., $A^c_{ij} \geq 0 \ \forall i \neq j$, then*
$$\overline{\lambda}_1(\boldsymbol{A}^S) = \lambda_1(\overline{A}).$$
2. *If A^Δ is diagonal, then*
$$\overline{\lambda}_1(\boldsymbol{A}^S) = \lambda_1(\overline{A}), \quad \underline{\lambda}_n(\boldsymbol{A}^S) = \lambda_n(\underline{A}).$$

Theorem

We have
$$\boldsymbol{\lambda}_i(\boldsymbol{A}^S) \subseteq [\lambda_i(A^c) - \rho(A^\Delta), \lambda_i(A^c) + \rho(A^\Delta)], \quad i = 1, \ldots, n.$$

Lower Bounds: αBB algorithm

Special cases: bilinear terms

For every $y \in \boldsymbol{y} \in \mathbb{IR}$ and $z \in \boldsymbol{z} \in \mathbb{IR}$ we have

$$yz \geq \max\{\underline{y}z + \underline{z}y - \underline{y}\underline{z},\ \overline{y}z + \overline{z}y - \overline{y}\overline{z}\}.$$

αBB algorithm (Androulakis, Maranas & Floudas, 1995)

- Consider an underestimator $g(x) \leq f(x)$ in the form

$$g(x) := f(x) + \alpha(x - \underline{x})^T(x - \overline{x}), \quad \text{where } \alpha \geq 0.$$

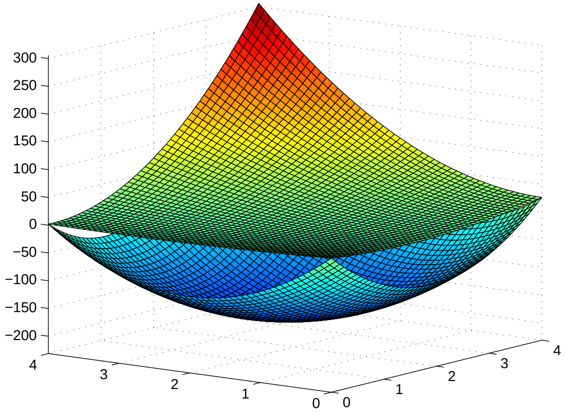
- We want $g(x)$ to be convex to easily determine $\underline{g}(\boldsymbol{x}) \leq \underline{f}(\boldsymbol{x})$.
- In order that $g(x)$ is convex, its Hessian

$$\nabla^2 g(x) = \nabla^2 f(x) + 2\alpha I_n$$

must be positive semidefinite on $x \in \boldsymbol{x}$. Thus we put

$$\alpha := -\tfrac{1}{2} \cdot \underline{\lambda}_{\min}(\nabla^2 f(\boldsymbol{x})).$$

Illustration of a Convex Underestimator



Function $f(x)$ and its convex underestimator $g(x)$.

Examples

Example (The COPRIN examples, 2007, precision $\sim 10^{-6}$)

- tf12 (origin: COCONUT, solutions: 1, computation time: 60 s)

$$\begin{aligned} \min\ & x_1 + \tfrac{1}{2}x_2 + \tfrac{1}{3}x_3 \\ \text{s.t.}\ & -x_1 - \tfrac{i}{m}x_2 - (\tfrac{i}{m})^2 x_3 + \tan(\tfrac{i}{m}) \le 0, \quad i = 1, \dots, m \ (m = 101). \end{aligned}$$

- o32 (origin: COCONUT, solutions: 1, computation time: 2.04 s)

$$\min\ 37.293239x_1 + 0.8356891x_5x_1 + 5.3578547x_3^2 - 40792.141$$

$$\begin{aligned} \text{s.t.}\quad -0.0022053x_3x_5 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 6.665593 &\le 0, \\ -0.0022053x_3x_5 - 0.0056858x_2x_5 - 0.0006262x_1x_4 - 85.334407 &\le 0, \\ 0.0071317x_2x_5 + 0.0021813x_3^2 + 0.0029955x_1x_2 - 29.48751 &\le 0, \\ -0.0071317x_2x_5 - 0.0021813x_3^2 - 0.0029955x_1x_2 + 9.48751 &\le 0, \\ 0.0047026x_3x_5 + 0.0019085x_3x_4 + 0.0012547x_1x_3 - 15.699039 &\le 0, \\ -0.0047026x_3x_5 - 0.0019085x_3x_4 - 0.0012547x_1x_3 + 10.699039 &\le 0. \end{aligned}$$

- Rastrigin (origin: Myatt (2004), solutions: 1 (approx.), time: 2.07 s)

$$\min\ 10n + \textstyle\sum_{j=1}^{n}(x_j - 1)^2 - 10\cos(2\pi(x_j - 1))$$

where $n = 10$, $x_j \in [-5.12, 5.12]$.

References

C. A. Floudas and P. M. Pardalos, editors.
Encyclopedia of Optimization. 2nd ed.
Springer, New York, 2009.

E. R. Hansen and G. W. Walster.
Global Optimization Using Interval Analysis.
Marcel Dekker, New York, second edition, 2004.

R. B. Kearfott.
Rigorous Global Search: Continuous Problems.
Kluwer, Dordrecht, 1996.

A. Neumaier.
Complete search in continuous global optimization and constraint satisfaction.
Acta Numer., 13:271–369, 2004.

H. Ratschek and J. Rokne.
New Computer Methods for Global Optimization.
Wiley, Chichester, 2007.

Rigorous Global Optimization Software

- *GlobSol* (by R. Baker Kearfott), written in Fortran 95, open-source
  `http://interval.louisiana.edu/`
- *Alias* (by COPRIN team), A C++ library with Maple interface,
  `http://www-sop.inria.fr/coprin/logiciels/ALIAS/`
- *IBEX* (by G. Chabert, B. Neveu, J. Ninin and others),
  an open-source interval C++ library, `http://www.ibex-lib.org/`
- *COCONUT Environment*, open-source C++ classes
  `http://www.mat.univie.ac.at/~coconut/coconut-environment/`
- *GLOBAL* (by Tibor Csendes), for Matlab / Intlab, free for academic
  `http://www.inf.u-szeged.hu/~csendes/linkek_en.html`
- *PROFIL / BIAS* (by O. Knüppel et al.), free C++ class
  `http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html`

See also

- *C.A. Floudas* (`http://titan.princeton.edu/tools/`)
- *A. Neumaier* (`http://www.mat.univie.ac.at/~neum/glopt.html`)