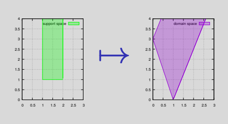
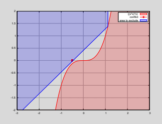
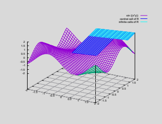
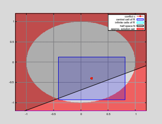
Generalizing Taylor models for multivariate real functions [×]









Mathematical Logic and its Applications, 22-24 March 2021, online

N. Th. Müller [1] M. Korovina [2] F. Brauße [3]

Abteilung Informatikwissenschaften, Universität Trier, Germany [1]

A.P. Ershov Institute of Informatics Systems, Novosibirsk, Russia [2]

The University of Manchester, UK [3]

[×] This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143 and from the DFG grant WERA MU 1801/5-1.

FroCoS'19, Brauße/Korovin/Korovina/Müller

Combine reliable real computations and resolution in a CDCL-style calculus called **KSMT** aiming at SMT solvers for nonlinear problems.

Main parts:

1. use CNF $\mathcal{C}$ in separated linear form $\mathcal{C} = \mathcal{L} \wedge \mathcal{N}$
   - $\mathcal{L}$ – clauses of linear inequalities: $q_1 x_1 + q_2 x_2 + \cdots + q_n x_n + q_0 \diamond 0$

$$\begin{array}{rrcl} & 2x_1 - 4x_2 - 2x_3 - 2 & > & 0 \\ \vee & 4x_2 + 2x_3 + 1 & \geq & 0 \end{array}$$

   - $\mathcal{N}$ – **unit clauses** of non-linear inequalities : $f(\overline{x}) \diamond y$

$$\begin{array}{rcl} x_2^2 \cdot x_1 & \leq & x_3 \\ sin(x_1^2 + x_2^2) & > & 0 \end{array}$$

2. conflict-driven calculus with well-known steps of
   - (R): linear inequality resolution
   - (B): backjumps
   - (A): assignment refinement
3. new step (L): local linearisations for resolving non-linear conflicts

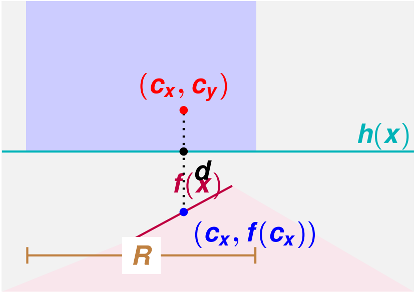
Constraint $P \equiv y \leq f(x)$, assignment $x \mapsto c_x$, $y \mapsto c_y$

Interval linearisation:

1. check for conflict
2. choose intermediate d

3. use constant function h with $h(c_x) = d$
4. and find suitable polytope(s) R with

$$y \leq f(x) \quad \Rightarrow \quad (\, x \notin R \vee y \leq h(x) \,)$$



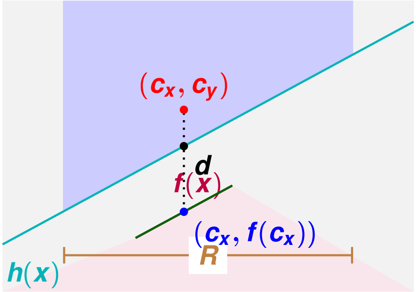
reasonable: R large with $R \subseteq \{x \mid h(x) \geq f(x)\}$

Constraint $P \equiv y \leq f(x)$, assignment $x \mapsto c_x$, $y \mapsto c_y$

Tangent space linearisation:

1. check for conflict
2. choose intermediate d
3. approximate slope ∂
4. choose linear function h with $h(c_x) = d$
5. and find suitable polytope(s) R with

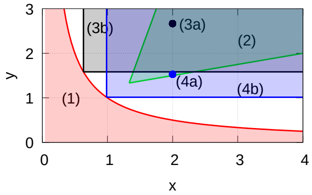
$$y \leq f(x) \quad \Rightarrow \quad (\ x \notin R \vee y \leq h(x)\ )$$



reasonable: R large with $R \subseteq \{x \mid h(x) \geq f(x)\}$

Example:

$$\mathcal{C} = (y \le 1/x)$$
$$\wedge (y \ge x/4 + 1)$$
$$\wedge (y \le 4 \cdot (x - 1))$$
$$\wedge ((x \le \tfrac{12}{19}) \vee (y \le \tfrac{19}{12}))$$
$$\wedge ((x \le \tfrac{220}{223}) \vee (y \le \tfrac{223}{220}))$$



Linearisation of $(y \le 1/x)$ and conflict (c_x, c_y):

- use $d := (c_y + 1/c_x)/2$
- $(y \le 1/x) \rightarrow$

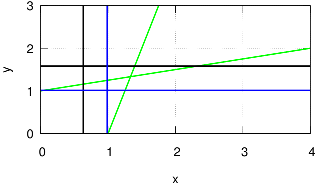
$$((x \le 1/d) \vee (y \le d))$$

rule	α	note
(A)	$x \mapsto 2$	
(A)	$x \mapsto 2,\ y \mapsto \frac{8}{3}$	(3a)
(L)	$x \mapsto 2,\ y \mapsto \frac{8}{3}$	(3b)
(B)	$x \mapsto 2$	
(A)	$x \mapsto 2,\ y \mapsto \frac{84}{55}$	(4a)
(L)	$x \mapsto 2,\ y \mapsto \frac{84}{55}$	(4b)
	...	
n/a		**unsat**

Interpretation:

$$\mathcal{C} = (y \leq 1/x)$$
$$\land (y \geq x/4 + 1)$$
$$\land (y \leq 4 \cdot (x - 1))$$

$$\land ((x \leq \tfrac{12}{19}) \lor (y \leq \tfrac{19}{12}))$$
$$\land ((x \leq \tfrac{220}{223}) \lor (y \leq \tfrac{223}{220}))$$



- lin. predicates $\sim$ prop. variables

$$\mathcal{C} = N$$
$$\land L_1$$
$$\land L_2$$
$$\land (L_3 \lor L_4)$$
$$\land (L_5 \lor L_6)$$

- Prop. variables:
  $\{N, L_1, L_2\} \bigcup \{L_3, L_4, L_5, L_6\}$
- linearisations $\sim$ clauses
  $(L_3 \lor L_4)$, $(L_5 \lor L_6)$
- parallelity $\sim$ implications
  $(\neg L_3 \lor L_5)$, $(\neg L_6 \lor L_4)$

- adding propositional variables is expensive
- adding clauses is quite cheap

Requirements (simplified...) for computing linearisations:

- input:
  - function $\boldsymbol{f}$, defined by term $\boldsymbol{t}$ using basic functions
  - vector $\boldsymbol{c_x} \in \mathbb{Q}^{\boldsymbol{d}}$ and value $\boldsymbol{c_y} \in \mathbb{Q}$ with $\boldsymbol{c_y} > \boldsymbol{f}(\boldsymbol{c_x})$
  - set $\boldsymbol{G} = \{\boldsymbol{g_1}, \boldsymbol{g_2}, \ldots, \boldsymbol{g_n}\}$ of linear functions

    defining initial polytope $\boldsymbol{R} := \bigcap_{i \leq n} \{\overline{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{d}} \mid \boldsymbol{g_i}(\overline{\boldsymbol{x}}) > \boldsymbol{0}\}$ with $\boldsymbol{c_x} \in \boldsymbol{R}$
- output:
  - function $\boldsymbol{h}$ and set $\boldsymbol{G'} = \{\boldsymbol{g'_1}, \boldsymbol{g'_2}, \ldots, \boldsymbol{g'_m}\}$ (all linear)

    defining a polytope $\boldsymbol{R'} := \bigcap_{i \leq m} \{\overline{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{d}} \mid \boldsymbol{g'_i}(\overline{\boldsymbol{x}}) > \boldsymbol{0}\}$ with $\boldsymbol{c_x} \in \boldsymbol{R'}$

    and

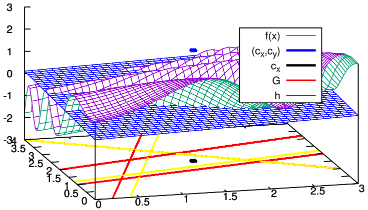
    $$\boldsymbol{c_y} > \boldsymbol{h}(\boldsymbol{c_x}), \quad \forall \overline{\boldsymbol{x}} \in \boldsymbol{R'} \quad \boldsymbol{h}(\overline{\boldsymbol{x}}) > \boldsymbol{f}(\overline{\boldsymbol{x}})$$

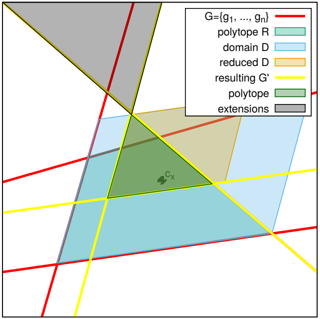
    and (possibly) additional clauses for cells $\boldsymbol{R''}$ satsfying
    $\forall \overline{\boldsymbol{x}} \in \boldsymbol{R''} \quad \boldsymbol{h}(\overline{\boldsymbol{x}}) > \boldsymbol{f}(\overline{\boldsymbol{x}})$
- input and output in symbolical form, only using rational numbers
- preferably with ‘large’ overlap $\boldsymbol{G'} \cap \boldsymbol{G}$

Basic algorithm

- Given: $\boldsymbol{f}$, $\boldsymbol{c_x}$, $\boldsymbol{c_y}$, $\boldsymbol{G}$
- $\boldsymbol{R} := \bigcap_{i \leq n} \{ \overline{\boldsymbol{x}} : \boldsymbol{g_i}(\overline{\boldsymbol{x}}) > \boldsymbol{0} \}$
- construct 'simple' $\boldsymbol{D}$ with $\boldsymbol{c_x} \in \boldsymbol{D}$
- repeat
  - compute linear function $\boldsymbol{h}$ with $(\forall \overline{\boldsymbol{x}} \in \boldsymbol{D}) \;\; \boldsymbol{h}(\overline{\boldsymbol{x}}) > \boldsymbol{f}(\overline{\boldsymbol{x}})$
  - if $\boldsymbol{c_y} \leq \boldsymbol{h}(\boldsymbol{c_x})$, shrink $\boldsymbol{D}$, keeping $\boldsymbol{c_x} \in \boldsymbol{D}$

  until $\boldsymbol{c_y} > \boldsymbol{h}(\boldsymbol{c_x})$
- match $\boldsymbol{D}$ with $\{ \boldsymbol{g_1}, \boldsymbol{g_2}, \ldots, \boldsymbol{g_n} \}$
- return $\boldsymbol{h}$ and $\boldsymbol{G'} = \{ \boldsymbol{g'_1}, \ldots \}$
- return polytopes with $\boldsymbol{h}(\overline{\boldsymbol{x}}) > \boldsymbol{f}(\overline{\boldsymbol{x}})$





Definition 2.1

- $[\boldsymbol{a}{:}\boldsymbol{b}]$ *is the closed interval* $\{\boldsymbol{z} \in \mathbb{R} \mid \boldsymbol{a} \leq \boldsymbol{z} \leq \boldsymbol{b}\}$
- $[\boldsymbol{a}]$ *is the point interval* $\{\boldsymbol{a}\} = [\boldsymbol{a}{:}\boldsymbol{a}]$
- *Infinite intervals:* $[\boldsymbol{a}{:}\infty]$, $[-\infty{:}\boldsymbol{b}]$,$[-\infty{:}\infty]$
- *Let* $\mathbb{K}$ *be a subset of* $\mathbb{R}$*, e.g.* $\mathbb{Q}, \mathbb{R}_{\boldsymbol{c}}, \mathbb{D}$*.*

  The extended real intervals $\mathbb{IK}$ *with bounds from* $\mathbb{K}$ *are*

$$\begin{aligned} \mathbb{IK} = \quad & \{\, [\boldsymbol{a}{:}\boldsymbol{b}] \mid \boldsymbol{a}, \boldsymbol{b} \in \mathbb{K}, \boldsymbol{a} \leq \boldsymbol{b} \,\} \\ \cup \quad & \{\, [\boldsymbol{a} : \infty] \mid \boldsymbol{a} \in \mathbb{K}\} \\ \cup \quad & \{\, [-\infty : \boldsymbol{b}] \mid \boldsymbol{b} \in \mathbb{K} \,\} \\ \cup \quad & \{\, [-\infty : \infty] \,\} \end{aligned}$$

Usual operations (perhaps approximate) on intervals:

- standard arithmetic $(+, -, \cdot, /)$, flags for finiteness
- inf, sup, mid, rad, sgn
- constructing point intervals, rational approximations

Valid laws for $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{IK}$, $\boldsymbol{a} \in \mathbb{K}$, if $\mathbb{K}$ is a ring:

- $\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$ and $\boldsymbol{A} \cdot \boldsymbol{B} = \boldsymbol{B} \cdot \boldsymbol{A}$, commutative
- $\boldsymbol{A} \subseteq \boldsymbol{C}; \boldsymbol{B} \subseteq \boldsymbol{D} \Rightarrow \boldsymbol{A} \circ \boldsymbol{B} \subseteq \boldsymbol{C} \circ \boldsymbol{D}$ inclusion monotone
- $(\boldsymbol{A} + \boldsymbol{B}) \cdot \boldsymbol{C} \subseteq \boldsymbol{A} \cdot \boldsymbol{C} + \boldsymbol{B} \cdot \boldsymbol{C}$, subdistributive
- $(\boldsymbol{A} + [\boldsymbol{b}]) \cdot [\boldsymbol{c}] = \boldsymbol{A} \cdot [\boldsymbol{c}] + [\boldsymbol{b}] \cdot \boldsymbol{c}$, distributive for point intervals

main task: Given $\boldsymbol{f}$ and $\overline{\boldsymbol{x}}$, find

- linear function $\boldsymbol{h}$ and polytope $\boldsymbol{D}$ with $\overline{\boldsymbol{x}} \in \boldsymbol{D}$
- with $\forall \overline{\boldsymbol{x}} \in \boldsymbol{D} : \boldsymbol{h}(\overline{\boldsymbol{x}}) > \boldsymbol{f}(\overline{\boldsymbol{x}})$

intermediate goal:

- data type for approximations of functions on polytopes
- interface to data type:
  - pointwise application of important real operators like ADD, SUB, MULT, SIN, ...

$$\begin{aligned} \mathrm{ADD}(\boldsymbol{f}, \boldsymbol{g}) = \boldsymbol{h} \quad &\rightsquigarrow \forall \overline{\boldsymbol{x}} \in \boldsymbol{D} : \boldsymbol{h}(\overline{\boldsymbol{x}}) = \boldsymbol{f}(\overline{\boldsymbol{x}}) + \boldsymbol{g}(\overline{\boldsymbol{x}}) \\ \mathrm{SIN}(\boldsymbol{f}) = \boldsymbol{h} \quad &\rightsquigarrow \forall \overline{\boldsymbol{x}} \in \boldsymbol{D} : \boldsymbol{h}(\overline{\boldsymbol{x}}) = \sin(\boldsymbol{f}(\overline{\boldsymbol{x}})) \end{aligned}$$

  - (approximative) evaluation at real arguments
  - linear bounds on (possibly infinite) polytopes

($\rightsquigarrow$ Representations in TTE sense can be derived...)

Generalize the idea of Taylor models [Makino/Berz]:

- use multivariate polynomial approximations
- with intervals coefficients
- use vectors $\overline{\lambda} = (\lambda_1, \dots, \lambda_k)^{\mathrm{T}}$ of parameters $\lambda_i \in S_i$ ('error symbols' in the setting of Taylor models)

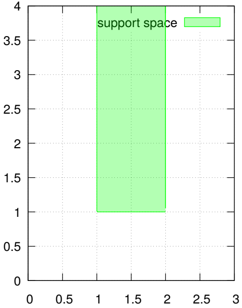
Modifications:

- use polynomials with (possibly infinite) interval coefficients
  (instead of a single finite interval for all errors)
- use rational / dyadic / computable real numbers
  (instead of double precision)
- use (possibly infinite) support space $S = S_1 \times \ldots \times S_k$
  (instead of 'usual' hypercube $[-1, 1]^k$)
- use parallelotopes as domain space D
  via a linear transformation Δ applied to S
  (instead of axes-aligned boxes)

Example: Consider

$$\boldsymbol{S} = \begin{pmatrix} [\mathbf{1} : \mathbf{2}] \\ [\mathbf{1} : \infty) \end{pmatrix}, \boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} \\ -\mathbf{3} & \mathbf{5} \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} -\mathbf{3} \\ \mathbf{1} \end{pmatrix}, \boldsymbol{A}^{-1} = \begin{pmatrix} \frac{\mathbf{5}}{\mathbf{11}} & \frac{-\mathbf{2}}{\mathbf{11}} \\ \frac{\mathbf{3}}{\mathbf{11}} & \frac{\mathbf{1}}{\mathbf{11}} \end{pmatrix}$$

support $\boldsymbol{S}$ and domain $\boldsymbol{D}$ are as follows:



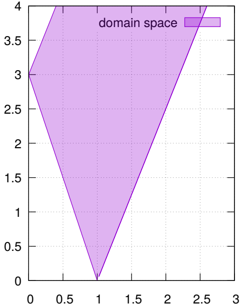
$$\Delta(\overline{\lambda}) = \boldsymbol{A} \cdot \overline{\lambda} + \boldsymbol{B}$$

$$\longrightarrow$$

$$\boldsymbol{S} \ni \overline{\lambda} \qquad\qquad \overline{\boldsymbol{x}} \in \boldsymbol{D}$$

$$\longleftarrow$$

$$\Delta^{-\mathbf{1}}(\overline{\boldsymbol{x}}) = \boldsymbol{A}^{-\mathbf{1}} \cdot (\overline{\boldsymbol{x}} - \boldsymbol{B})$$



Definition 3.2

- *A Taylor model base (TMB)* $\boldsymbol{T}$ *consists of the following components:*
  - *a column vector* $\overline{\boldsymbol{S}} = (\boldsymbol{S}_1, \ldots, \boldsymbol{S}_k)^{\mathrm{T}}$ *with* $\boldsymbol{S}_i \in \mathbb{IK}$
  - *an invertible affine transformation* Δ*,*
    given by a nonsingular matrix $\boldsymbol{A} \in \mathbb{K}^{k \times k}$ *and a vector* $\boldsymbol{B} \in \mathbb{K}^k$
- *The support space* $\boldsymbol{S_T} \subseteq \mathbb{R}^k$ *of* $\boldsymbol{T}$ *is the cartesian product*

$$\boldsymbol{S_T} = \bigtimes \boldsymbol{S_i}$$

- *The domain space* $\boldsymbol{D_T} \subseteq \mathbb{R}^k$ *of* $\boldsymbol{T}$ *is the set*

$$\boldsymbol{D_T} = \{\boldsymbol{A} \cdot \overline{\lambda} + \boldsymbol{B} \mid \overline{\lambda} \in \boldsymbol{S_T}\}$$

Definition 3.3

- *Consider multivariate polynomials **p** in $\overline{\lambda}$:*

$$\boldsymbol{p}(\overline{\lambda}) = \sum_{\overline{\boldsymbol{n}}} \boldsymbol{c}_{\overline{\boldsymbol{n}}} \cdot \overline{\lambda}^{\overline{\boldsymbol{n}}}$$

*where **p** is given by its coefficients $\boldsymbol{c}_{\overline{\boldsymbol{n}}} \in \mathbb{IK}$.*

- *A function **f** : $\subseteq\mathbb{R}^{\boldsymbol{k}} \to \mathbb{R}$ is approximated by **p** w.r.t. **T**, iff*

$$\boldsymbol{D_T} \subseteq \mathbf{dom}(\boldsymbol{f})$$

and

$$\forall \overline{\boldsymbol{x}} \in \boldsymbol{D_T} \quad \boldsymbol{f}(\overline{\boldsymbol{x}}) \in \mathbf{p}(\Delta^{-1}(\overline{\boldsymbol{x}}))$$

(in symbols: $\boldsymbol{f} \sim^{\boldsymbol{T}} \boldsymbol{p}$)

Convergence of the approximations:

Lemma 3.4 (Weierstraß approximation theorem)

If $\boldsymbol{D_T}$ is compact, every continuous $\boldsymbol{f} : \boldsymbol{D_T} \to \mathbb{R}$ can be approximated arbitrarily precise by polynomials.

Lemma 3.5 (Interval arithmetic)

For every continuous $\boldsymbol{f} : \mathbb{R}^{\boldsymbol{k}} \to \mathbb{R}$, every $\overline{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{k}}$ and for any precision, there are $\boldsymbol{T}$ and a linear $\boldsymbol{p}$ with $\overline{\boldsymbol{x}} \in \boldsymbol{D_T}$ approximating $\boldsymbol{f}$ on $\boldsymbol{D_T}$ with the desired precision.

So, choose between extremes of high order polynomials or small domains...

Interval oriented operations: Replace ocurrence of λ_i in $\boldsymbol{p}$ by $\boldsymbol{s_i}$, e.g.

- Sweep: Replace $\overline{\lambda}^{\overline{n}}$ within $\boldsymbol{p}$ by $\overline{\boldsymbol{s}}^{\overline{n}}$:

$$\ldots + \underbrace{\boldsymbol{c}}_{\text{interval}} \cdot \underbrace{\overline{\lambda}^{\overline{n}} \cdot \overline{\lambda}^{\overline{m}}}_{\text{monomial}} + \ldots \quad \rightsquigarrow \quad \ldots + \underbrace{\boldsymbol{c} \cdot \overline{\boldsymbol{S}}^{\overline{n}}}_{\text{interval}} \cdot \underbrace{\overline{\lambda}^{\overline{m}}}_{\text{monomial}} + \ldots$$

- Reduce: Sweep *all* monomials $\overline{\lambda}^{\overline{n}}$ in $\boldsymbol{p}$ with $\overline{\boldsymbol{n}} \neq \overline{\boldsymbol{0}}$ in $\boldsymbol{p}$.

$$\boldsymbol{p} \quad \rightsquigarrow \quad \boldsymbol{p}(\begin{pmatrix} \boldsymbol{S_1} \\ \vdots \\ \boldsymbol{S_k} \end{pmatrix}) \quad \in \mathbb{IK}$$

- Pointifying: For arbitrary *point* $\boldsymbol{d}$ apply

$$\ldots + \underbrace{\boldsymbol{c}}_{\text{interval}} \cdot \underbrace{\overline{\lambda}^{\overline{n}}}_{\text{monomial}} + \ldots \quad \rightsquigarrow \quad \ldots + \underbrace{\boldsymbol{d}}_{\text{point}} \cdot \underbrace{\overline{\lambda}^{\overline{n}}}_{\text{monomial}} + \underbrace{(\boldsymbol{c} - \boldsymbol{d}) \cdot \boldsymbol{S}^{\overline{n}}}_{\text{interval}} + \ldots$$

Further operations on the Taylor models, e.g.

- Splitting: Replace interval coefficient $\boldsymbol{c}$ by a *new* variable $\lambda_{\boldsymbol{c}}$ with support $\boldsymbol{c}$, e.g.

$$\ldots + \underbrace{\boldsymbol{c}}_{\text{interval}} \cdot \underbrace{\overline{\lambda}^{\overline{\boldsymbol{n}}}}_{\text{monomial}} + \ldots \quad \rightsquigarrow \quad \ldots + \underbrace{[\mathbf{1}]}_{\text{interval}} \cdot \underbrace{\lambda_{\boldsymbol{c}} \cdot \overline{\lambda}^{\overline{\boldsymbol{n}}}}_{\text{monomial}} + \ldots$$

  $\lambda_{\boldsymbol{c}}$: shortcut for this occurrence of interval $\boldsymbol{c}$ in further computations, without changes to domain and support spaces!

- Square sweep: Preferably replace $\lambda_{\boldsymbol{i}}^{\mathbf{2}}$ by $\boldsymbol{S}_{\boldsymbol{i}}^{\mathbf{2}} \subseteq \mathbb{R}_{\mathbf{0}}^{+}$ (instead of $\boldsymbol{S}_{\boldsymbol{i}} \cdot \boldsymbol{S}_{\boldsymbol{i}} \subseteq \mathbb{R}$) in case $\mathbf{0} \in \boldsymbol{S}_{\boldsymbol{i}}$.

Lemma 3.6

Suppose $\boldsymbol{p} \rightsquigarrow \boldsymbol{p}'$ *via sweeping/reducing/pointifying/splitting , then*

$$\boldsymbol{f} \sim^{\boldsymbol{T}} \boldsymbol{p} \quad \Longrightarrow \quad \boldsymbol{f} \sim^{\boldsymbol{T}} \boldsymbol{p}'$$

- Reducing $\boldsymbol{p}$ leads to a single interval $\boldsymbol{c} \in \mathbb{IK}$.
  So if $\mathbf{0} \notin \boldsymbol{c}$, then $\boldsymbol{f}(\overline{\boldsymbol{x}}) \neq \mathbf{0}$ on $\boldsymbol{D_T}$.
- Sweep and pointify $\boldsymbol{p}$ leading to

$$\boldsymbol{p}' = \boldsymbol{c} + \overline{\boldsymbol{d}}_{\overline{\boldsymbol{n}}} \cdot \overline{\lambda}^{\overline{\boldsymbol{n}}}$$

with *interval* $\boldsymbol{c}$ and *point vectors* $\overline{\boldsymbol{d}}_{\overline{\boldsymbol{n}}}$.
Then, if $\boldsymbol{c}$ has finite upper bound $\sup(\boldsymbol{c})$,

$$\boldsymbol{h}(\overline{\boldsymbol{x}}) := \sup(\boldsymbol{c}) + \overline{\boldsymbol{d}}_{\overline{\boldsymbol{n}}} \cdot (\Delta^{-1}(\overline{\boldsymbol{x}}))^{\overline{\boldsymbol{n}}}$$

is a *polynomial* with

$$\forall \overline{\boldsymbol{x}} \in \boldsymbol{D_T} \quad \boldsymbol{h}(\overline{\boldsymbol{x}}) \geq \boldsymbol{f}(\overline{\boldsymbol{x}})$$

(similar for lower bound $\inf(\boldsymbol{c})$)

Example 3.7

The projections $\mathbf{pr}_{\boldsymbol{i}} : \boldsymbol{D_T} \to \mathbb{R}, \begin{pmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_k} \end{pmatrix} \mapsto \boldsymbol{x_i}$ *are approximated by*

$$\overline{\boldsymbol{pr_i}} : \boldsymbol{S_T} \to \mathbb{R}, \quad \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} \mapsto \boldsymbol{e_i} \cdot \left(\boldsymbol{A} \cdot \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} + \boldsymbol{B}\right)$$

where $\boldsymbol{e_i} = (\underbrace{[\mathbf{0}], \ldots, [\mathbf{0}], [\mathbf{1}]}_{\boldsymbol{i}}, [\mathbf{0}], \ldots, [\mathbf{0}])$ *is the* $\boldsymbol{i}$*-th unit vector consisting of point intervals.*

In example with $\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} \\ -\mathbf{3} & \mathbf{5} \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} -\mathbf{3} \\ \mathbf{1} \end{pmatrix}$

$$\begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{pmatrix} \mapsto \boldsymbol{x_1} \quad \sim^{\boldsymbol{T}} \quad \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \mapsto [\mathbf{1}] \cdot \lambda_1 + [\mathbf{2}] \cdot \lambda_2 + [-\mathbf{3}]$$

$$\begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{pmatrix} \mapsto \boldsymbol{x_2} \quad \sim^{\boldsymbol{T}} \quad \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \mapsto [-\mathbf{3}] \cdot \lambda_1 + [\mathbf{5}] \cdot \lambda_2 + [\mathbf{1}]$$

Example 3.8

Let $f_1 \sim^T p_1$ *and* $f_2 \sim^T p_2$.

Let $\begin{Bmatrix} \circ \\ \diamond \end{Bmatrix}$ *be* $\begin{Bmatrix} \text{pointwise} \\ \text{formal} \end{Bmatrix}$ *product of* $\begin{Bmatrix} \text{functions} \\ \text{polynomials} \end{Bmatrix}$.

Then $f_1 \circ f_2 \sim^T p_1 \diamond p_2$. *The same holds for sum and difference.*

In example with $A = \begin{pmatrix} 1 & 2 \\ -3 & 5 \end{pmatrix}$, $B = \begin{pmatrix} -3 \\ 1 \end{pmatrix}$

$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto x_1 \cdot x_2$ is approximated by

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \mapsto ([1]\lambda_1 + [2]\lambda_2 + [-3]) \cdot ([-3]\lambda_1 + [5]\lambda_2 + [1])$$
$$= [-3]\lambda_1^2 + [-1]\lambda_1\lambda_2 + [10]\lambda_2^2 + [10]\lambda_1 + [-13]\lambda_2 + [-3]$$

Consider example $\boldsymbol{f}(\boldsymbol{x_1}, \boldsymbol{x_2}) = \boldsymbol{x_1} \cdot \boldsymbol{x_2}$ with TMB as above, using $\boldsymbol{S_1} = [\mathbf{1}{:}\mathbf{2}]$ and $\boldsymbol{S_2} = [\mathbf{1}{:}\infty]$:

$$
\begin{aligned}
& [-\mathbf{3}]\lambda_1^2 + [-\mathbf{1}]\lambda_1\lambda_2 + [\mathbf{10}]\lambda_2^2 + [\mathbf{10}]\lambda_1 + [-\mathbf{13}]\lambda_2 + [-\mathbf{3}] \\
\overset{\text{sweep}}{\rightsquigarrow}\quad & [-\mathbf{3}] \cdot [\mathbf{1}{:}\mathbf{2}] \cdot \lambda_1 + [-\mathbf{1}][\mathbf{1}{:}\mathbf{2}] \cdot \lambda_2 + [\mathbf{10}] \cdot [\mathbf{1}{:}\infty] \cdot \lambda_2 \\
& \qquad + [\mathbf{10}]\lambda_1 + [-\mathbf{13}]\lambda_2 + [-\mathbf{3}] \\
= \quad & [\mathbf{4}{:}\mathbf{7}] \cdot \lambda_1 + [-\mathbf{5}{:}\infty] \cdot \lambda_2 + [-\mathbf{3}] \\
\overset{\text{pointify}}{\rightsquigarrow}\quad & [\mathbf{0}{:}\mathbf{3}] \cdot \lambda_1 + [\mathbf{0}{:}\infty] \cdot \lambda_2 + [\mathbf{4}]\lambda_1 + [-\mathbf{5}]\lambda_2 + [-\mathbf{3}] \\
\overset{\text{sweep}}{\rightsquigarrow}\quad & [\mathbf{0}{:}\mathbf{6}] \qquad + [\mathbf{0}{:}\infty] \qquad + [\mathbf{4}]\lambda_1 + [-\mathbf{5}]\lambda_2 + [-\mathbf{3}]
\end{aligned}
$$

So consider

$$\boldsymbol{h} : \boldsymbol{S_T} \to \mathbb{R}, \qquad \boldsymbol{h}(\lambda_1, \lambda_2) := \mathbf{4}\lambda_1 - \mathbf{5}\lambda_2 - \mathbf{3}$$

after transforming back to $\boldsymbol{D_T}$ with $\overline{\lambda} = \boldsymbol{A}^{-1} \cdot (\overline{\boldsymbol{x}} - \boldsymbol{B})$ to $\boldsymbol{D_T}$

$$\forall \overline{\boldsymbol{x}} \in \boldsymbol{D_T} \qquad \boldsymbol{x_1} \cdot \boldsymbol{x_2} \geq \frac{\mathbf{5} \cdot \boldsymbol{x_1} - \mathbf{13} \cdot \boldsymbol{x_2} - \mathbf{5}}{\mathbf{11}}$$

`tangentspace:`

- standalone linearizer
- prototypical implementation in `C++`
- ca 400 SLOC: small user interface, plotting
- ca 400 SLOC: core routines (`tangentspace.*`)
- ca 1000 SLOC: helper routines, Taylor models (`tm4ts.*`)
- using rational numbers from `GMP`
- currently supported basic functions:

  `+`, `-`, `*`, `/`, `sin`, `cos`, `exp`, `abs`, `sgn`

Usage: From standard input read strings like:

- `function - * x0 x0 sin x1 end`

  (define multivariate function $\boldsymbol{f}(\boldsymbol{x_0}, \boldsymbol{x_1}) = \boldsymbol{x_0^2} - \boldsymbol{sin}(\boldsymbol{x_1})$ *using prefix notation)*

- `args -7/10 2/10 end`

  (Specify argument vector $\boldsymbol{c_x}$*, here* $(\boldsymbol{c_x} = (\mathbf{0.7}, \mathbf{0.2}$ *)*

- `conflict 3/10`

  (Specify a 'conflict' $\boldsymbol{c_y}$*,usually with* $\boldsymbol{c_y} \neq \boldsymbol{f}(\boldsymbol{c_x})$ *)*

- `bounds 0 100/100 -60/100 end`

  (Define lists of coordinates for recycling,

  for variable $\boldsymbol{x_0}$ *try to use bounds* $\frac{\mathbf{100}}{\mathbf{100}} = \mathbf{1}$ *and* $\frac{\mathbf{-60}}{\mathbf{100}} = \mathbf{-0.6}$ *)*

- `effort 20`

  (Restrict the precision in the linearization process

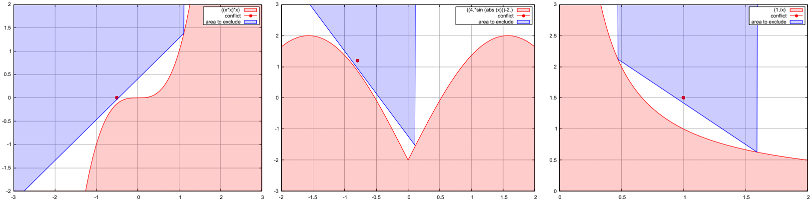
  to radii larger than $\mathbf{2^{-20}} \rightsquigarrow$ *failure for* $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x_0}, \boldsymbol{x_1})$ *)*

- `plot`

  (Give gnuplot commands and run them, plot area via `area`*, 2d- or 3d-plot)*

Example inputs for `tangentspace`

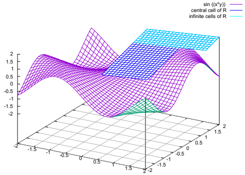
- predicates with two variables
- $\boldsymbol{f}(\boldsymbol{x_0}) \diamond \boldsymbol{y}$
- 2d-plot

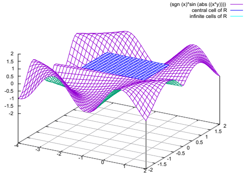


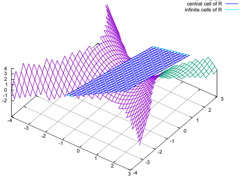
1. `function * * x0 x0 x0 end args -5/10 end conflict 0/1 area -3 3 -2 2 end show end plot`
2. `function - * 4 sin abs x0 2 end args -8/10 end conflict 12/10 ratio 10 1 area -2 2 -3 3 end plot end`
3. `function / 1 x0 end args 1 end conflict 3/2 area 0 2 0 3 end plot end`

Example inputs for `tangentspace`

- predicates with three variables
- $\boldsymbol{f}(\boldsymbol{x_0}, \boldsymbol{x_1}) \diamond \boldsymbol{y}$
- 3d-plot



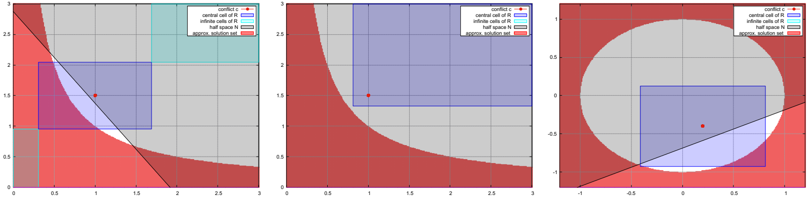




1. `function sin * x0 x1 end args 1/2 1/5 end  conflict 1/1 area -2 2 -2 2 end plot end`
2. `function * sgn x0 sin abs * x0 x1 end args 0 0 end conflict 1/1 area -2 2 -2 2 end plot end`
3. `function - * x0 x0 * x1 x1 end args 1/3 1/3 end  conflict 2/1 area -4 3 -4 3 -2 4 end plot end`

Example inputs for `tangentspace`,

- predicates with two variables, no c_y
- $f(x_0, x_1) \diamond 0$
- 2d-plot



1. `function - * x0 x1 1  end args 1 3/2 end`
   `area 0 3 0 3 end plot end`
2. `function - * x0 x1 1  end args 1 3/2 end`
   `interval area 0 3 0 3 end plot end`
3. `function - + * x0 x0 * x1 x1 1  end  args 1/5 -2/5 end`
   `area -1.2 1.2 -1.2 1.2 end plot end`

Current state:

- currently: linear models with rational coefficients
- under construction: higher order models with computable coefficients

ToDo:

- implementing affine transformation
- optimization for cells
- integration in SMT solver
- composition, limits, ...

Thank you for your attention!

Questions? Remarks?