Subdivision Algorithms and the CF Expansion of Real Roots of Polynomial Systems

Angelos Mantzaflaris, Bernard Mourrain and Elias Tsigaridas

GALAAD

INRIA Sophia-Antipolis – Méditerranée

Algorithms Project's Seminar, September 28, 2009

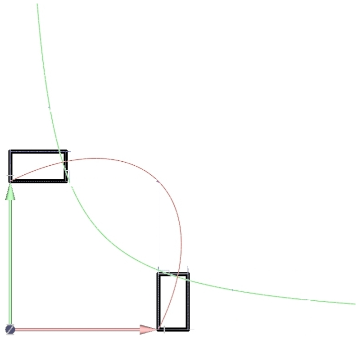
Real Root isolation

Example

$$f = x^5 - 7x^4 + 22x^3 - 4x^2 - 48x + 36 = (x-1)\cdot(x^2-6x+18)\cdot(x^2-2)$$

real roots	$-\sqrt{2}$	1	$+\sqrt{2}$
output	$(-49, 0)$	$(\frac{49}{64}, \frac{147}{128})$	$(\frac{147}{128}, 49)$

- 0-dim systems of polynomial equations, e.g. $\{f_1, f_2\} \subseteq \mathbb{Z}[x, y]$

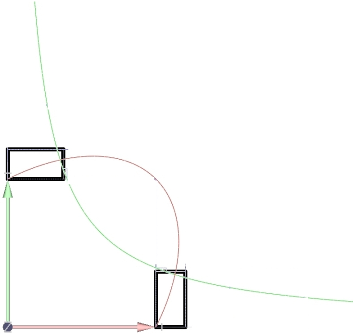


Example

$$f = x^5 - 7x^4 + 22x^3 - 4x^2 - 48x + 36 = (x-1)\cdot(x^2-6x+18)\cdot(x^2-2)$$

real roots	$-\sqrt{2}$	1	$+\sqrt{2}$
output	$(-49, 0)$	$(\frac{49}{64}, \frac{147}{128})$	$(\frac{147}{128}, 49)$

- 0-dim systems of polynomial equations, e.g. $\{f_1, f_2\} \subseteq \mathbb{Z}[x, y]$



Continued fractions

Any $\zeta \in \mathbb{R}$ can be written as

$$\zeta = b_0 + \cfrac{1}{b_1 + \cfrac{1}{\ddots}} = [b_0, b_1, b_2, \ldots]$$

Example

$$\sqrt{8} = 2 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cfrac{1}{\ddots}}}} = [2, 1, 4, 1, \ldots]$$

- Partial approximant of bitsize τ yields the best rational τ-approximation of the real number

Continued fractions

Any $\zeta \in \mathbb{R}$ can be written as

$$\zeta = b_0 + \cfrac{1}{b_1 + \cfrac{1}{\ddots}} = [b_0, b_1, b_2, \ldots]$$

Example

$$\sqrt{8} = 2 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cfrac{1}{\ddots}}}} = [2, 1, 4, 1, \ldots]$$

- Partial approximant of bitsize τ yields the best rational τ-approximation of the real number

Continued fractions

Any $\zeta \in \mathbb{R}$ can be written as

$$\zeta = b_0 + \cfrac{1}{b_1 + \cfrac{1}{\ddots}} = [b_0, b_1, b_2, \dots]$$

Example

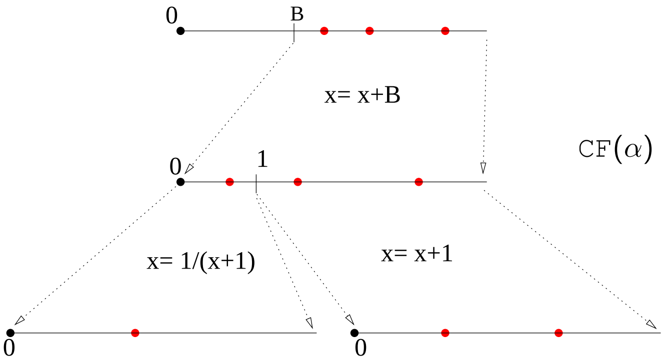
$$\sqrt{8} = 2 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cfrac{1}{\ddots}}}} = [2, 1, 4, 1, \dots]$$

- Partial approximant of bitsize τ yields the best rational τ-approximation of the real number

CF Algorithm for Real Root Isolation

To isolate the positive real roots of $f \in \mathbb{Z}[x]$:

- Compute a positive integer lower bound B, reduce domain (0,B]
- Check for **one** or **no** solution by *Descartes' rule of signs*
- Subdivide using *Homography transformations* and repeat..



$$\mathtt{CF}(\alpha) = \lfloor \alpha \rfloor + \frac{1}{\mathtt{CF}\left(\frac{1}{\alpha - \lfloor \alpha \rfloor}\right)}$$

CF: Termination & Complexity

Theorem ([Vincent;1836], [Uspensky;1948], [Alesina,Galuzzi;1998])

Let $f \in \mathbb{Z}[x]$, and $b_0, b_1, \ldots, b_n \in \mathbb{Z}_+$, $n > \mathcal{O}(d\,\tau)$. The map

$$x \mapsto b_0 + \cfrac{1}{b_1 + \cfrac{1}{\ddots\, b_n + \cfrac{1}{x}}}$$

transforms $f(x)$ to $\tilde{f}(x)$ such that

1. $V(F) = 0 \Leftrightarrow f$ has no positive real roots.
2. $V(F) = 1 \Leftrightarrow f$ has one positive real root.

Average complexity [Tsigaridas, Emiris; 2008]

The expected complexity of **CF** is $\widetilde{\mathcal{O}}_B(d^3\tau)$.

CF: Termination & Complexity

Theorem ([Vincent;1836], [Uspensky;1948], [Alesina,Galuzzi;1998])

Let $f \in \mathbb{Z}[x]$, and $b_0, b_1, \ldots, b_n \in \mathbb{Z}_+$, $\quad n > \mathcal{O}(d\,\tau)$. The map

$$x \mapsto b_0 + \cfrac{1}{b_1 + \cfrac{1}{\ddots\, b_n + \cfrac{1}{x}}}$$

transforms $f(x)$ to $\tilde{f}(x)$ such that

1. $V(F) = 0 \Leftrightarrow f$ has no positive real roots.
2. $V(F) = 1 \Leftrightarrow f$ has one positive real root.

Average complexity [Tsigaridas, Emiris; 2008]

The expected complexity of **CF** is $\widetilde{\mathcal{O}}_B(d^3\tau)$.

A General Subdivision Scheme

Input. A system $f_1, f_2, .., f_s \in \mathbb{Z}[\mathbf{x}]$ represented over a domain $\mathcal{I}$.
Output. A list of disjoint domains, each containing one and only one real root of $f_1 = \cdots = f_s = 0$.

Initialize a stack Q and add $(\mathcal{I}, f_1, .., f_s)$ on top of it
While Q is not empty do

a) Pop a system $(\mathcal{I}, f_1, .., f_s)$ and:
b) Perform a precondition process and/or a reduction process to refine the domain.
c) Apply an exclusion test to identify if the domain contains no roots. Apply an inclusion test to identify if the domain contains a single root. In this case output $(\mathcal{I}, f_1, .., f_s)$.
d) If both tests fail split the representation over $\mathcal{I}$ into a number of sub-domains and push them to Q.

A General Subdivision Scheme

Input. A system $f_1, f_2, .., f_s \in \mathbb{Z}[\mathbf{x}]$ represented over a domain $\mathcal{I}$.
Output. A list of disjoint domains, each containing one and only one real root of $f_1 = \cdots = f_s = 0$.
Initialize a stack Q and add $(\mathcal{I}, f_1, .., f_s)$ on top of it
While Q is not empty do

a) Pop a system $(\mathcal{I}, f_1, .., f_s)$ and:
b) Perform a precondition process and/or a reduction process to refine the domain.
c) Apply an exclusion test to identify if the domain contains no roots. Apply an inclusion test to identify if the domain contains a single root. In this case output $(\mathcal{I}, f_1, .., f_s)$.
d) If both tests fail split the representation over $\mathcal{I}$ into a number of sub-domains and push them to Q.

A General Subdivision Scheme

Input. A system $f_1, f_2, .., f_s \in \mathbb{Z}[\mathbf{x}]$ represented over a domain $\mathcal{I}$.
Output. A list of disjoint domains, each containing one and only one real root of $f_1 = \cdots = f_s = 0$.
Initialize a stack Q and add $(\mathcal{I}, f_1, .., f_s)$ on top of it
While Q is not empty do

a) Pop a system $(\mathcal{I}, f_1, .., f_s)$ and:
b) Perform a precondition process and/or a reduction process to refine the domain.
c) Apply an exclusion test to identify if the domain contains no roots. Apply an inclusion test to identify if the domain contains a single root. In this case output $(\mathcal{I}, f_1, .., f_s)$.
d) If both tests fail split the representation over $\mathcal{I}$ into a number of sub-domains and push them to Q.

A General Subdivision Scheme

Input. A system $f_1, f_2, .., f_s \in \mathbb{Z}[\mathbf{x}]$ represented over a domain $\mathcal{I}$.
Output. A list of disjoint domains, each containing one and only one real root of $f_1 = \cdots = f_s = 0$.
Initialize a stack Q and add $(\mathcal{I}, f_1, .., f_s)$ on top of it
While Q is not empty do

a) Pop a system $(\mathcal{I}, f_1, .., f_s)$ and:
b) Perform a precondition process and/or a reduction process to refine the domain.
c) Apply an exclusion test to identify if the domain contains no roots. Apply an inclusion test to identify if the domain contains a single root. In this case output $(\mathcal{I}, f_1, .., f_s)$.
d) If both tests fail split the representation over $\mathcal{I}$ into a number of sub-domains and push them to Q.

A General Subdivision Scheme

Input. A system $f_1, f_2, .., f_s \in \mathbb{Z}[\mathbf{x}]$ represented over a domain $\mathcal{I}$.
Output. A list of disjoint domains, each containing one and only one real root of $f_1 = \cdots = f_s = 0$.
Initialize a stack Q and add $(\mathcal{I}, f_1, .., f_s)$ on top of it
While Q is not empty do

a) Pop a system $(\mathcal{I}, f_1, .., f_s)$ and:
b) Perform a precondition process and/or a reduction process to refine the domain.
c) Apply an exclusion test to identify if the domain contains no roots. Apply an inclusion test to identify if the domain contains a single root. In this case output $(\mathcal{I}, f_1, .., f_s)$.
d) If both tests fail split the representation over $\mathcal{I}$ into a number of sub-domains and push them to Q.

A General Subdivision Scheme

Input. A system $f_1, f_2, .., f_s \in \mathbb{Z}[\mathbf{x}]$ represented over a domain $\mathcal{I}$.
Output. A list of disjoint domains, each containing one and only one real root of $f_1 = \cdots = f_s = 0$.
Initialize a stack Q and add $(\mathcal{I}, f_1, .., f_s)$ on top of it
While Q is not empty do

a) Pop a system $(\mathcal{I}, f_1, .., f_s)$ and:
b) Perform a precondition process and/or a reduction process to refine the domain.
c) Apply an exclusion test to identify if the domain contains no roots. Apply an inclusion test to identify if the domain contains a single root. In this case output $(\mathcal{I}, f_1, .., f_s)$.
d) If both tests fail split the representation over $\mathcal{I}$ into a number of sub-domains and push them to Q.

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

The **mCF** algorithm

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

The **mCF** algorithm

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

The **mCF** algorithm

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

The **mCF** algorithm

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

The **mCF** algorithm

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

The **mCF** algorithm

- Representation by homography. Subdivision using Taylor shifts
- Reduction using univariate projections, preconditioning using the Jacobian.
- two criteria: identify a single solution in domain (inclusion)
  identify a domain with no solutions (exclusion)

Consequently:

- works in monomial basis
- uses only integer arithmetic
- treats unbounded domains
- computes CF expansion (i.e. best rational approximations) of the coordinates of the roots

Overview

Representation

Homography (or Möbius transformation)

Bijective projective transformation $\mathcal{H} = (\mathcal{H}_1, .., \mathcal{H}_n)$ over $\mathbb{P}^1 \times \cdots \times \mathbb{P}^1$,

$$x_k \mapsto \mathcal{H}_k(x_k) = \frac{\alpha_k x_k + \beta_k}{\gamma_k x_k + \delta_k} \;, \quad \alpha_k, \beta_k, \gamma_k, \delta_k \in \mathbb{Z}, \;\; \gamma_k \delta_k \neq 0, \;\; k = 1, .., n$$

$$H(f) := \prod_{k=1}^{n} (\gamma_k x_k + \delta_k)^{d_k} \cdot (f \circ \mathcal{H})(x)$$

Base homographies:

- translation by $c \in \mathbb{Z}$: $T_k^c(f) = f|_{x_k = x_k + c}$
- contraction by $c \in \mathbb{Z}$: $C_k^c(f) = f|_{x_k = c x_k}$
- reciprocal polynomial: $R_k(f) = x_k^{d_k} f|_{x_k = 1/x_k}$

Lemma

The group of homographies is generated by R_k, C_k^c, T_k^c, $k = 1, .., n$.

Representation

Homography (or Möbius transformation)

Bijective projective transformation $\mathcal{H} = (\mathcal{H}_1, .., \mathcal{H}_n)$ over $\mathbb{P}^1 \times \cdots \times \mathbb{P}^1$,

$$x_k \mapsto \mathcal{H}_k(x_k) = \frac{\alpha_k x_k + \beta_k}{\gamma_k x_k + \delta_k} \ , \quad \alpha_k, \beta_k, \gamma_k, \delta_k \in \mathbb{Z}, \ \ \gamma_k \delta_k \neq 0, \ \ k = 1, .., n$$

$$H(f) := \prod_{k=1}^{n} (\gamma_k x_k + \delta_k)^{d_k} \cdot (f \circ \mathcal{H})(x)$$

Base homographies:

- translation by $c \in \mathbb{Z}$: $T_k^c(f) = f|_{x_k = x_k + c}$
- contraction by $c \in \mathbb{Z}$: $C_k^c(f) = f|_{x_k = c x_k}$
- reciprocal polynomial: $R_k(f) = x_k^{d_k} f|_{x_k = 1/x_k}$

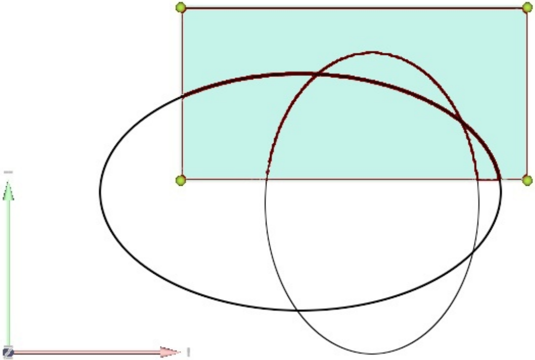
Lemma

The group of homographies is generated by R_k, C_k^c, T_k^c, $k = 1, .., n$.

Representation

$$\mathcal{H} = (x, y)$$

$$\boldsymbol{f}$$



Initial system $\boldsymbol{f} = (f_1, f_2)$ of two ellipses.
We compute a homography representation over the box

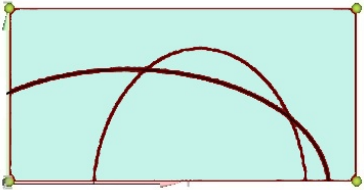
$$[1, 3] \times [1, 2]$$

Representation

$$\mathcal{H} = (x+1, y+1)$$

$$T_1^1 \, T_2^1$$

$$(\boldsymbol{f})$$



Translate both variables by 1 *(using Horner's scheme)*.

$$H(f) = (f \circ \mathcal{H})(x, y)$$

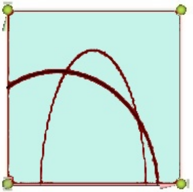
Representation

$$\mathcal{H} = (2x + 1, y + 1)$$

C_1^2

$T_1^1 \, T_2^1$

$(\boldsymbol{f})$



Contract $x-$variable by a factor of 2 *(multiply coeff. of $x^i y^j$ by 2^i)*.

$$H(f) = (f \circ \mathcal{H})(x, y)$$

Representation

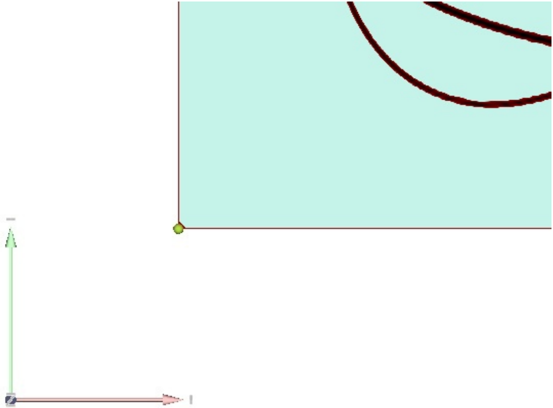
$$\mathcal{H} = \left(\frac{x+2}{x}, \frac{y+1}{y}\right)$$

$$R_1 R_2$$

$$C_1^2$$

$$T_1^1 T_2^1$$

$$(\boldsymbol{f})$$



Invert both variables *(swap coefficients $x^i y^j$ and $x^{2-i} y^{2-j}$)*.

$$H(f) = x^2 y^2 (f \circ \mathcal{H})(x, y)$$

Representation

$$\mathcal{H} = \left(\frac{x+3}{x+1}, \frac{y+2}{y+1}\right)$$

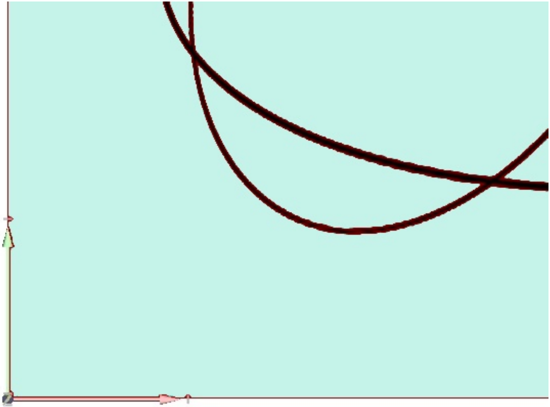
$T_1^1\, T_2^1$

$R_1\, R_2$

C_1^2

$T_1^1\, T_2^1$

$(\boldsymbol{f})$



Translate both variables by 1. Now $\mathcal{H}(\mathbb{R}_+^2) = [1,3] \times [1,2]$.

$$H(f) = (x+1)^2(y+1)^2(f \circ \mathcal{H})(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2}\binom{2}{i}\binom{2}{j} b_{2-i,2-j}\cdot x^i y^j$$

Representation

$$\mathcal{H} = \left(\frac{x+3}{x+1}, \frac{y+2}{y+1}\right)$$

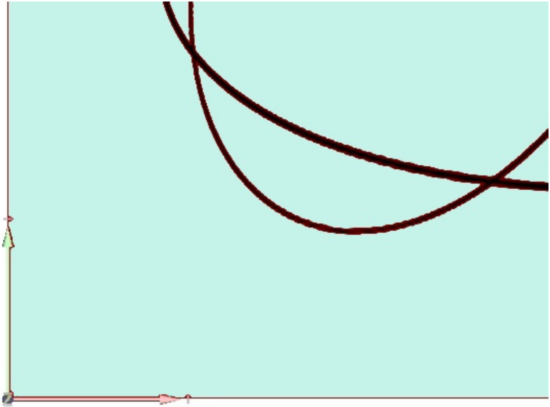
$$T_1^1 \, T_2^1$$

$$R_1 \, R_2$$

$$C_1^2$$

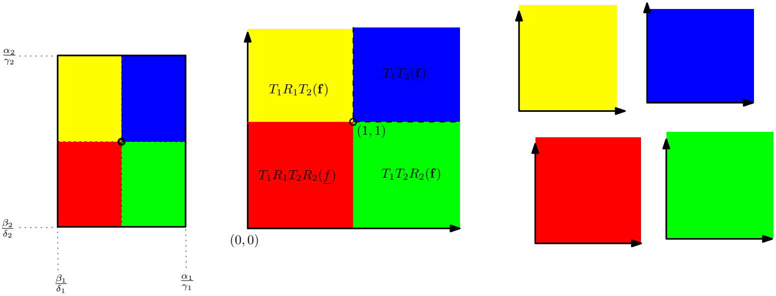
$$T_1^1 \, T_2^1$$

$$(\boldsymbol{f})$$



Translate both variables by 1. Now $\mathcal{H}(\mathbb{R}_+^2) = [1,3] \times [1,2]$.

$$H(f) = (x+1)^2(y+1)^2(f \circ \mathcal{H})(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2}\binom{2}{i}\binom{2}{j} b_{2-i,2-j} \cdot x^i y^j$$

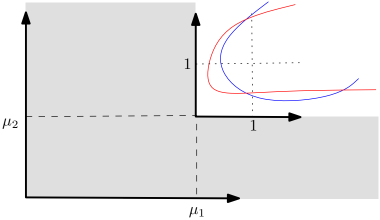


Keep in memory:

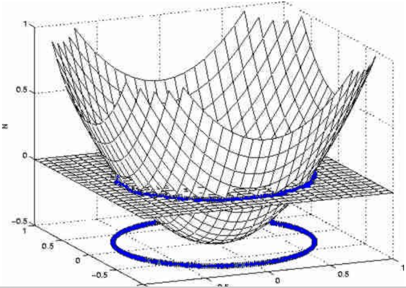
- Transformed polynomials: $H(f_1), \ldots, H(f_s)$ as coefficient *tensors*.
- $4n$ integers: $\alpha_k, \beta_k, \gamma_k, \delta_k$, $k = 1, \ldots, n$ to keep track of the domain.

Reduction Step

- Reducing the domain using lower bounds

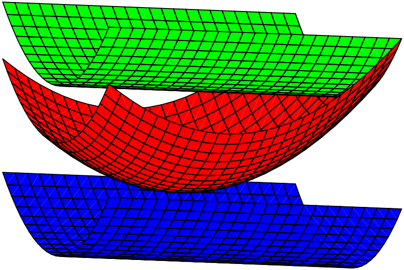


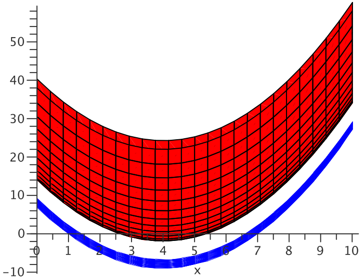
- The graph of f_i in $\mathbb{R}^{n+1}$



Reduction Step

$$m_k(f;x_k) = \sum_{i_k=0}^{d_k} \min_{i_1,..,\widehat{i_k},..,i_n} c_{i_1..i_n} x_k^{i_k} \quad , \quad M_k(f;x_k) = \sum_{i_k=0}^{d_k} \max_{i_1,..,\widehat{i_k},..,i_n} c_{i_1..i_n} x_k^{i_k}$$





- Following ideas of Bernstein algorithm [Mourrain,Pavone'09]

Reduction Step

$$m_k(f;x_k) = \sum_{i_k=0}^{d_k} \min_{i_1,..,\widehat{i_k},..,i_n} c_{i_1..i_n} x_k^{i_k} \quad , \quad M_k(f;x_k) = \sum_{i_k=0}^{d_k} \max_{i_1,..,\widehat{i_k},..,i_n} c_{i_1..i_n} x_k^{i_k}$$

Lemma

$$m_k(f;x_k) \leq \frac{f(x)}{\prod_{s\neq k} \sum_{i_s=0}^{d_s} x_s^{i_s}} \leq M_k(f;x_k) \quad , \quad k = 1,..,n$$

Corollary (lower bounds on the coordinates of the zeros)

$$\mu_k := \begin{cases} \text{min. pos. root of } M_k(f,x_k) & \text{if } M_k(f;0) < 0 \\ \text{min. pos. root of } m_k(f,x_k) & \text{if } m_k(f;0) > 0 \\ 0 & \text{otherwise} \end{cases}$$

All positive roots of f lie in $\mathbb{R}_{>\mu_1} \times \cdots \times \mathbb{R}_{>\mu_n}$.

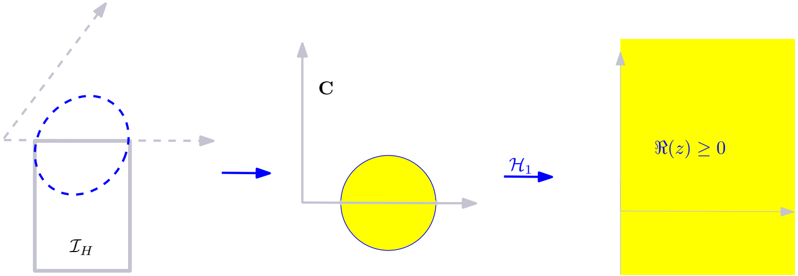
Exclusion Criterion

Vincent Theorem in several variables

Let $f(\boldsymbol{x}) = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{d}} c_{\boldsymbol{i}}\, \boldsymbol{x}^{\boldsymbol{i}}$ with $c_{\boldsymbol{i}} \in \mathbb{R}$, without (complex) solutions s.t. $\Re(z_k) \geq 0$ for some k. Then all its coefficients $c_{\boldsymbol{i}}$ are of the same sign.

Corollary

If the complex multidisk associated to a domain $\mathcal{I}_H$ does not intersect $\{z \in (\mathbb{P}^1)^n \;:\; f_i(z) = 0\} \Rightarrow$ the coeffs. of $H(f_i)$ have no sign changes.



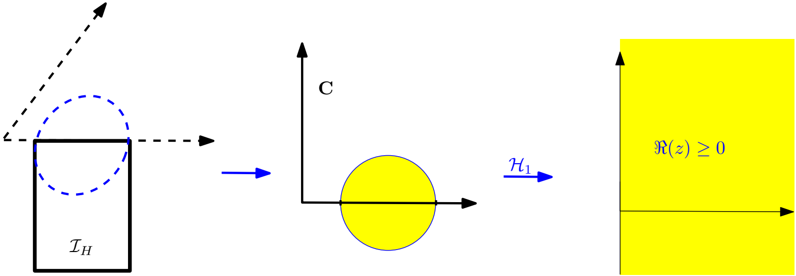
Exclusion Criterion

Vincent Theorem in several variables

Let $f(\boldsymbol{x}) = \sum_{i=0}^{d} c_i\, \boldsymbol{x}^i$ with $c_i \in \mathbb{R}$, without (complex) solutions s.t. $\Re(z_k) \geq 0$ for some k. Then all its coefficients c_i are of the same sign.

Corollary

If the complex multidisk associated to a domain $\mathcal{I}_H$ does not intersect $\{z \in (\mathbb{P}^1)^n \;:\; f_i(z) = 0\} \Rightarrow$ the coeffs. of $H(f_i)$ have no sign changes.



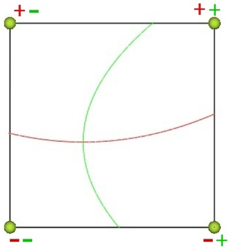
Inclusion Criterion

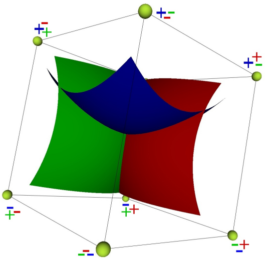
Miranda Theorem

If for every pair of parallel faces there exists f_i that attains opposite signs on the faces, then $f_1, .., f_n$ have at least one root inside the box.

Lemma

If the Jacobian has a constant sign in the box, then there is at most one root of $f_1, .., f_n$ inside the box.





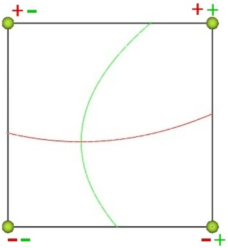
Inclusion Criterion

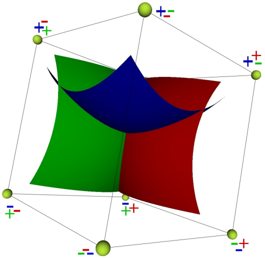
Miranda Theorem

If for every pair of parallel faces there exists f_i that attains opposite signs on the faces, then $f_1, .., f_n$ have at least one root inside the box.

Lemma

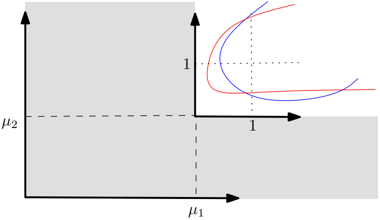
If the Jacobian has a constant sign in the box, then there is at most one root of $f_1, .., f_n$ inside the box.





Complexity

$$\zeta_1 = \mu_1^{(0)} + \cfrac{1}{\mu_1^{(1)} + \cfrac{1}{\mu_1^{(2)} + \cdots}} = \frac{P_{k_i(\zeta)}}{Q_{k_i(\zeta)}}$$



- $\Delta_i(\zeta)$: local separation bound of ζ_i,
  $k_i(\zeta)$: # of steps that isolate ζ_i

$$\left| \frac{P_{k_i(\zeta)}}{Q_{k_i(\zeta)}} - \zeta_i \right| < \phi^{-2k_i(\zeta)+1} \leq \Delta_i(\zeta),$$

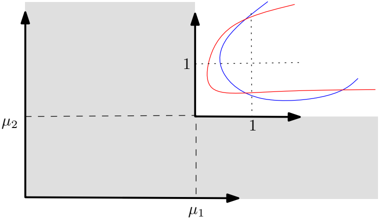
- Generalization of DMM bound:

$$\prod_{\zeta \in V} \Delta_i(\zeta) \geq 2^{-2n\tau d^{2n-1} - d^{2n}/2} \, (nd^n)^{-nd^{2n}}$$

[Emiris,Mourrain,Tsigaridas]

Complexity

$$\zeta_1 = \mu_1^{(0)} + \cfrac{1}{\mu_1^{(1)} + \cfrac{1}{\mu_1^{(2)} + \cdots}} = \frac{P_{k_i(\zeta)}}{Q_{k_i(\zeta)}}$$



- $\Delta_i(\zeta)$: local separation bound of ζ_i,
  $k_i(\zeta)$: # of steps that isolate ζ_i

$$\left| \frac{P_{k_i(\zeta)}}{Q_{k_i(\zeta)}} - \zeta_i \right| < \phi^{-2k_i(\zeta)+1} \leq \Delta_i(\zeta),$$

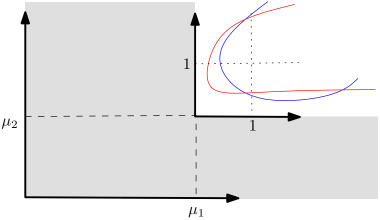
- Generalization of DMM bound:

$$\prod_{\zeta \in V} \Delta_i(\zeta) \geq 2^{-2n\tau d^{2n-1} - d^{2n}/2} \left(nd^n\right)^{-nd^{2n}}$$

[Emiris,Mourrain,Tsigaridas]

Complexity

$$\zeta_1 = \mu_1^{(0)} + \cfrac{1}{\mu_1^{(1)} + \cfrac{1}{\mu_1^{(2)} + \cdots}} = \frac{P_{k_i(\zeta)}}{Q_{k_i(\zeta)}}$$



- $\Delta_i(\zeta)$: local separation bound of ζ_i,
  $k_i(\zeta)$: # of steps that isolate ζ_i

$$\left| \frac{P_{k_i(\zeta)}}{Q_{k_i(\zeta)}} - \zeta_i \right| < \phi^{-2k_i(\zeta)+1} \leq \Delta_i(\zeta),$$

- Generalization of DMM bound:

$$\prod_{\zeta \in V} \Delta_i(\zeta) \geq 2^{-2n\tau d^{2n-1} - d^{2n}/2} (nd^n)^{-nd^{2n}}$$

[Emiris,Mourrain,Tsigaridas]

Continued

Two assumptions:

- The `include()` and `exclude()` tests always give a correct answer.
- The computed lower bound μ_k is optimal,
  i.e. coincides with the partial quotient of the CF expansion.
- Overall

$$\#\text{STEPS} \leq n \sum_{\zeta \in V} k_i(\zeta) \leq n\frac{1}{2}R - n\frac{1}{2}\sum_{\zeta \in V} \lg \Delta_i(\zeta)$$

$$\leq 2n\tau d^{2n-1} + 2nd^n \lg(nd^{2n})$$

Lemma

*The number of reduction/subdivision steps of **mCF** is $\widetilde{\mathcal{O}}(n^2 \tau d^{2n-1})$.*

Continued

Two assumptions:

- The `include()` and `exclude()` tests always give a correct answer.
- The computed lower bound μ_k is optimal, i.e. coincides with the partial quotient of the CF expansion.

- Overall

$$\#\text{STEPS} \leq n \sum_{\zeta \in V} k_i(\zeta) \leq n\frac{1}{2} R - n\frac{1}{2} \sum_{\zeta \in V} \lg \Delta_i(\zeta)$$
$$\leq 2n\tau d^{2n-1} + 2nd^n \lg(nd^{2n})$$

Lemma

*The number of reduction/subdivision steps of **mCF** is $\widetilde{\mathcal{O}}(n^2 \tau d^{2n-1})$.*

Continued

Two assumptions:

- The `include()` and `exclude()` tests always give a correct answer.
- The computed lower bound μ_k is optimal,
  i.e. coincides with the partial quotient of the CF expansion.
- Overall

$$
\begin{aligned}
\#\text{STEPS} \leq & \quad n \sum_{\zeta \in V} k_i(\zeta) \leq n\frac{1}{2}R - n\frac{1}{2}\sum_{\zeta \in V} \lg \Delta_i(\zeta) \\
\leq & \quad 2n\tau d^{2n-1} + 2nd^n \lg(nd^{2n})
\end{aligned}
$$

Lemma

*The number of reduction/subdivision steps of **mCF** is $\widetilde{\mathcal{O}}(n^2 \tau d^{2n-1})$.*

Complexity

- Complexity of shifting ($\mathbf{x} = \mathbf{x} + \mathbf{u}$) [Gathen,,Gerhard;1997]: $\widetilde{\mathcal{O}}_B(n^2 d^n \tau + d^{n+1} n^3 \sigma)$, obtained as nd^{n-1} univariate shifts
- Bound computation with cost $\mathcal{C}_1$, Tests evaluation wiith cost $\mathcal{C}_2$

Theorem

The total complexity is $\widetilde{\mathcal{O}}_B(2^n n^7 d^{5n-1} \tau^2 \sigma + (\mathcal{C}_1 + \mathcal{C}_2) n^2 \tau d^{n-1})$.

- Best rational approximation of the (coords. of the) real roots
- $n = 1$: matches average complexity of [Tsigaridas, Emiris'08]

Improvement by initial scaling:
Apply $C_k^{1/2^\ell}$ to the input.
The real roots are multiplied by 2^ℓ and their distance increases.
Total complexity improves by an order of d^{2n}.

Complexity

- Complexity of shifting ($\mathbf{x} = \mathbf{x} + \mathbf{u}$) [Gathen,,Gerhard;1997]: $\widetilde{\mathcal{O}}_B(n^2 d^n \tau + d^{n+1} n^3 \sigma)$, obtained as nd^{n-1} univariate shifts
- Bound computation with cost $\mathcal{C}_1$, Tests evaluation wiith cost $\mathcal{C}_2$

Theorem

The total complexity is $\widetilde{\mathcal{O}}_B(2^n n^7 d^{5n-1} \tau^2 \sigma + (\mathcal{C}_1 + \mathcal{C}_2) n^2 \tau d^{n-1})$.

- Best rational approximation of the (coords. of the) real roots
- $n = 1$: matches average complexity of [Tsigaridas, Emiris'08]

Improvement by initial scaling:
Apply $C_k^{1/2^\ell}$ to the input.
The real roots are multiplied by 2^ℓ and their distance increases.
Total complexity improves by an order of d^{2n}.

Complexity

- Complexity of shifting ($\mathbf{x} = \mathbf{x} + \mathbf{u}$) [Gathen,,Gerhard;1997]: $\widetilde{\mathcal{O}}_B(n^2 d^n \tau + d^{n+1} n^3 \sigma)$, obtained as nd^{n-1} univariate shifts
- Bound computation with cost $\mathcal{C}_1$, Tests evaluation wiith cost $\mathcal{C}_2$

Theorem

The total complexity is $\widetilde{\mathcal{O}}_B(2^n n^7 d^{5n-1} \tau^2 \sigma + (\mathcal{C}_1 + \mathcal{C}_2) n^2 \tau d^{n-1})$.

- Best rational approximation of the (coords. of the) real roots
- $n = 1$: matches average complexity of [Tsigaridas, Emiris'08]

Improvement by initial scaling:
Apply $C_k^{1/2^\ell}$ to the input.
The real roots are multiplied by 2^ℓ and their distance increases.
Total complexity improves by an order of d^{2n}.

Implementation

- **mCF** is implemented in MATHEMAGIX, in the C++ module `realroot`.
- Uses GMP arithmetic to work with large integer coefficients.
- Polynomials based on tensor (higher dimensional matrix) representation. Shift operations performed in place.
- Univariate solving by classic CF algorithm, special case of **mCF**. DFS traversal of the subdivision tree returns only the (floor of the) first positive root.
- AXEL Algebraic Modeler used for visualization in small dimensions.
  Application to topology and arrangement computations of implicit curves/surfaces.

Implementation

- **mCF** is implemented in MATHEMAGIX, in the C++ module `realroot`.
- Uses GMP arithmetic to work with large integer coefficients.
- Polynomials based on tensor (higher dimensional matrix) representation. Shift operations performed in place.
- Univariate solving by classic CF algorithm, special case of **mCF**. DFS traversal of the subdivision tree returns only the (floor of the) first positive root.
- AXEL Algebraic Modeler used for visualization in small dimensions.
  Application to topology and arrangement computations of implicit curves/surfaces.

Implementation

- **mCF** is implemented in MATHEMAGIX, in the C++ module `realroot`.
- Uses GMP arithmetic to work with large integer coefficients.
- Polynomials based on tensor (higher dimensional matrix) representation. Shift operations performed in place.
- Univariate solving by classic CF algorithm, special case of **mCF**. DFS traversal of the subdivision tree returns only the (floor of the) first positive root.
- AXEL Algebraic Modeler used for visualization in small dimensions.
  Application to topology and arrangement computations of implicit curves/surfaces.

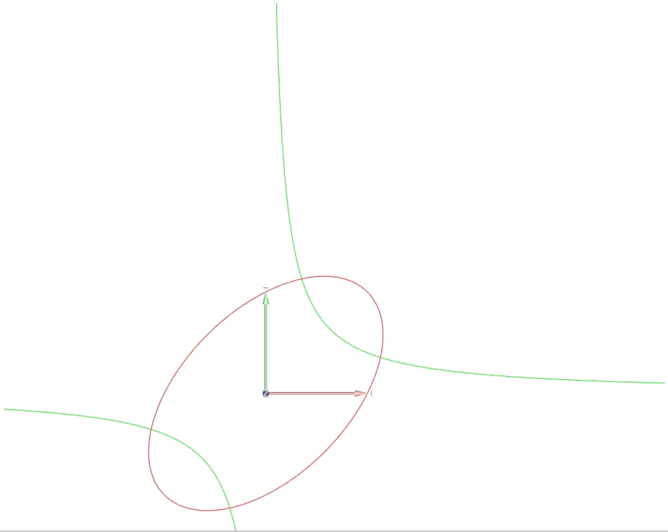
- **mCF** is implemented in MATHEMAGIX, in the C++ module `realroot`.
- Uses GMP arithmetic to work with large integer coefficients.
- Polynomials based on tensor (higher dimensional matrix) representation. Shift operations performed in place.
- Univariate solving by classic CF algorithm, special case of **mCF**. DFS traversal of the subdivision tree returns only the (floor of the) first positive root.
- AXEL Algebraic Modeler used for visualization in small dimensions.
  Application to topology and arrangement computations of implicit curves/surfaces.

Implementation

- **mCF** is implemented in MATHEMAGIX, in the C++ module `realroot`.
- Uses GMP arithmetic to work with large integer coefficients.
- Polynomials based on tensor (higher dimensional matrix) representation. Shift operations performed in place.
- Univariate solving by classic CF algorithm, special case of **mCF**. DFS traversal of the subdivision tree returns only the (floor of the) first positive root.
- AXEL Algebraic Modeler used for visualization in small dimensions.
  Application to topology and arrangement computations of implicit curves/surfaces.

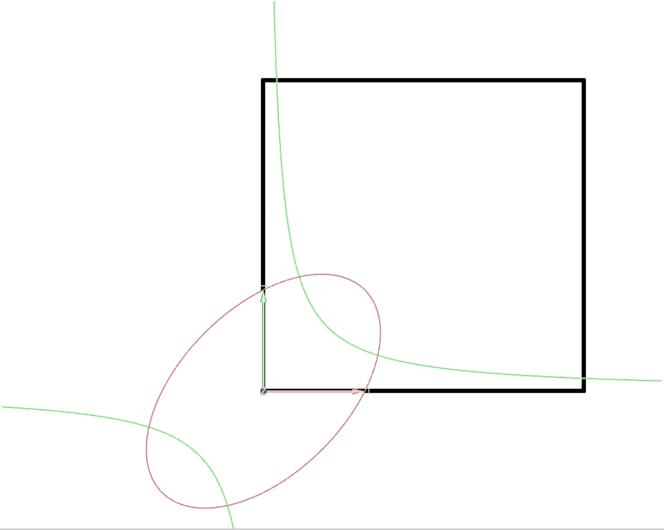
Toy example $\mathcal{I} = \mathbb{R}^2$

$$\boldsymbol{f}(x, y) = (f_1, f_2) = (y^2 - xy + x^2 - 1,\ 10xy - 4)$$



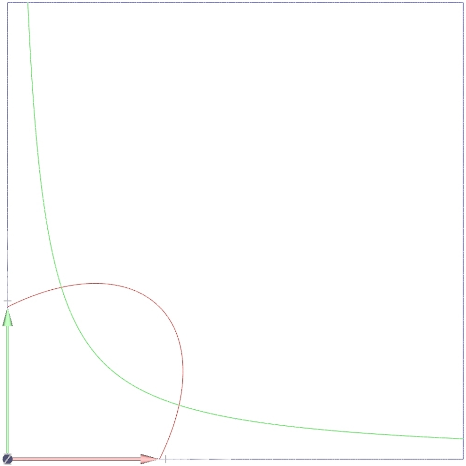
Toy example $\mathcal{I} = [0, 3] \times [0, 3]$

$$\boldsymbol{f}(x, y) \rightsquigarrow \boldsymbol{f}\left(\frac{3x}{x+1}, \frac{3y}{y+1}\right)$$



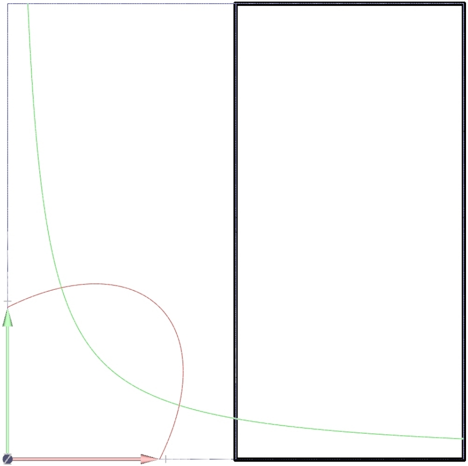
Toy example $\mathcal{I} = [0,3] \times [0,3]$

$$\boldsymbol{f}\left(\frac{3x}{x+1}, \frac{3y}{y+1}\right)$$



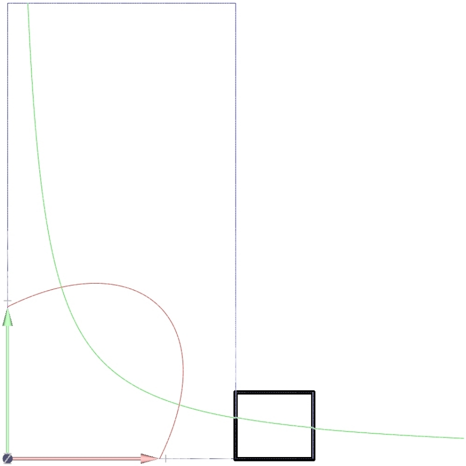
Toy example $\mathcal{I} = [\frac{3}{2}, 3] \times [0, 3]$

$$\boldsymbol{f}\left(\frac{3x+3}{x+2}, \frac{3y}{y+1}\right)$$



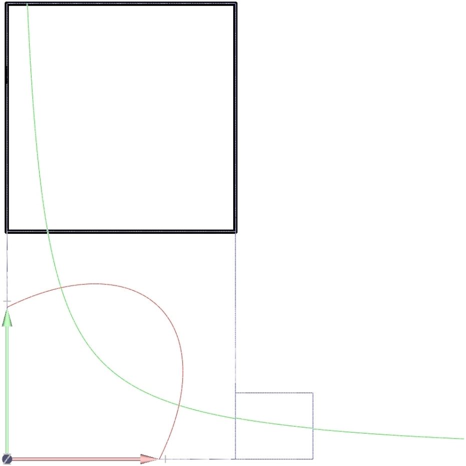
Toy example $\mathcal{I} = [\frac{3}{2}, 2] \times [0, \frac{3}{7}]$

$$\boldsymbol{f}\left(\frac{6x+3}{3x+2}, \frac{3y}{7y+1}\right)$$



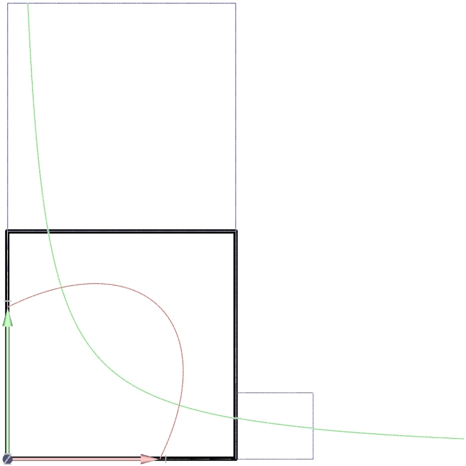
Toy example $\mathcal{I} = [0, \frac{3}{2}] \times [\frac{3}{2}, 3]$

$$\boldsymbol{f}\left(\frac{3x}{2x+1}, \frac{3y+3}{y+2}\right)$$



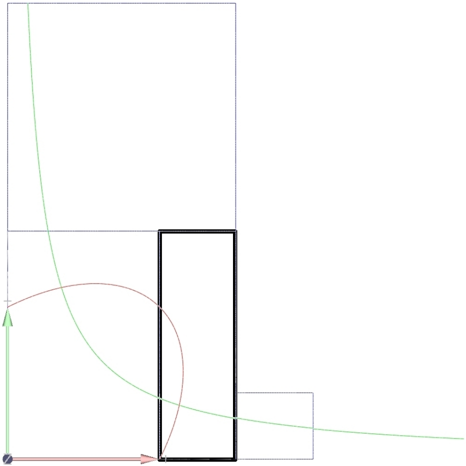
Toy example $\mathcal{I} = [0, \frac{3}{2}] \times [0, \frac{3}{2}]$

$$\boldsymbol{f}\left(\frac{3x}{2x+1}, \frac{3y}{2y+1}\right)$$



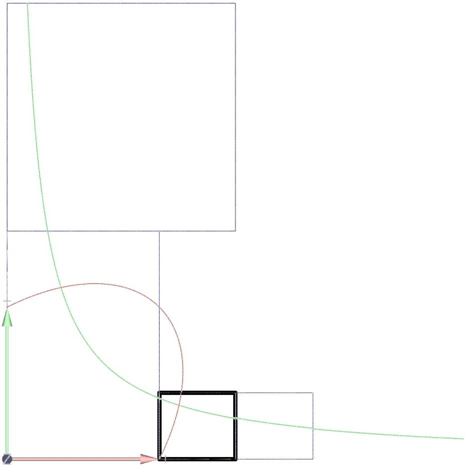
Toy example $\mathcal{I} = [1, \frac{3}{2}] \times [0, \frac{3}{2}]$

$$\boldsymbol{f}\left(\frac{3x+3}{2x+3}, \frac{3y}{2y+1}\right)$$



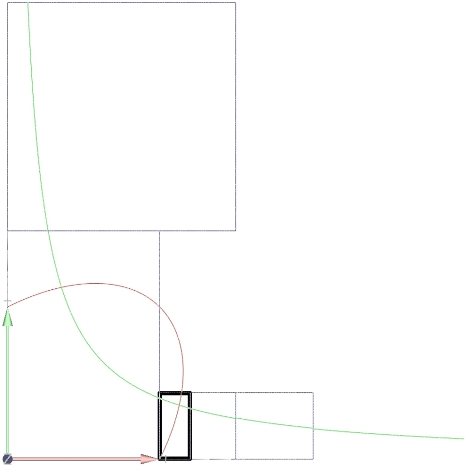
Toy example $\mathcal{I} = [1, \frac{3}{2}] \times [0, \frac{3}{7}]$

$$\boldsymbol{f}\left(\frac{3x+3}{2x+3}, \frac{3y}{7y+1}\right)$$



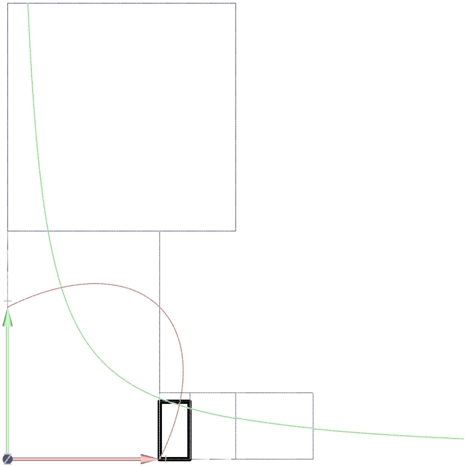
Toy example $\mathcal{I} = [1, \frac{6}{5}] \times [0, \frac{3}{7}]$

$$\boldsymbol{f}\left(\frac{6x+3}{5x+3}, \frac{3y}{7y+1}\right)$$



Toy example $\mathcal{I} = [1, \frac{6}{5}] \times [0, \frac{3}{8}]$

$$f\left(\frac{6x+3}{5x+3}, \frac{3y}{8y+1}\right)$$



Toy example $\mathcal{I}_1 = [1, \frac{6}{5}] \times [0, \frac{3}{8}]$ $\mathcal{I}_2 = [0, \frac{3}{8}] \times [1, \frac{6}{5}]$

$$\boldsymbol{f}\left(\frac{6x+3}{5x+3}, \frac{3y}{8y+1}\right) \qquad \boldsymbol{f}\left(\frac{3x}{8x+1}, \frac{6y+3}{5y+3}\right)$$