C++ application for hyperbolicity verification

Tomasz Kułaga

Joint work with:
M. Mazur, J. Tabor, P. Kościelniak
Jagiellonian University

Computational Hyperbolicity Group
http://www.im.uj.edu.pl/MarcinMazur/comphyp

DyToComp Conference, Będlewo 2009

Problem

Verifying hyperbolicity of non-trivial invariant set K for real (complex) Hénona map

$$H_{a,b}(x, y) = (a - x^2 + by, x)$$

for different parameter values $a, b \in \mathbb{R}$ (or $\mathbb{C}$)

We want to modify the problem in such a way that it would be possible to solve it using computer.

Problem

Verifying hyperbolicity of non-trivial invariant set K for real (complex) Hénona map

$$H_{a,b}(x, y) = (a - x^2 + by, x)$$

for different parameter values $a, b \in \mathbb{R}$ (or $\mathbb{C}$)

We want to modify the problem in such a way that it would be possible to solve it using computer.

Possible approaches

Some concepts of computational hyperbolicity:

- *Quasi-hyperbolicity* (Arai)

  Z. Arai, *On Hyperbolic Plateaus of the Hénon Maps*, Experiment. Math. 16 (2007), 181–188.

- *Cone fields* (Hruska)

  S. L. Hruska, *A Numerical Method for Constructing the Hyperbolic Structure of Complex Hénon Mappings*, Found. Comput. Math. 6 (2006), 427–455.

- *Semi-hyperbolicity* (Mazur, Tabor, Kościelniak, Kułaga)

  M. Mazur, J. Tabor, *Computational hyperbolicity*, preprint.

Possible approaches

Some concepts of computational hyperbolicity:

- *Quasi-hyperbolicity* (Arai)

  Z. Arai, *On Hyperbolic Plateaus of the Hénon Maps*, Experiment. Math. 16 (2007), 181–188.

- *Cone fields* (Hruska)

  S. L. Hruska, *A Numerical Method for Constructing the Hyperbolic Structure of Complex Hénon Mappings*, Found. Comput. Math. 6 (2006), 427–455.

- *Semi-hyperbolicity* (Mazur, Tabor, Kościelniak, Kułaga)

  M. Mazur, J. Tabor, *Computational hyperbolicity*, preprint.

Possible approaches

Some concepts of computational hyperbolicity:

- *Quasi-hyperbolicity* (Arai)

  Z. Arai, *On Hyperbolic Plateaus of the Hénon Maps*, Experiment. Math. 16 (2007), 181–188.

- *Cone fields* (Hruska)

  S. L. Hruska, *A Numerical Method for Constructing the Hyperbolic Structure of Complex Hénon Mappings*, Found. Comput. Math. 6 (2006), 427–455.

- *Semi-hyperbolicity* (Mazur, Tabor, Kościelniak, Kułaga)

  M. Mazur, J. Tabor, *Computational hyperbolicity*, preprint.

Possible approaches

Some concepts of computational hyperbolicity:

- *Quasi-hyperbolicity* (Arai)

  Z. Arai, *On Hyperbolic Plateaus of the Hénon Maps*, Experiment. Math. 16 (2007), 181–188.

- *Cone fields* (Hruska)

  S. L. Hruska, *A Numerical Method for Constructing the Hyperbolic Structure of Complex Hénon Mappings*, Found. Comput. Math. 6 (2006), 427–455.

- *Semi-hyperbolicity* (Mazur, Tabor, Kościelniak, Kułaga)

  M. Mazur, J. Tabor, *Computational hyperbolicity*, preprint.

Hyperbolicity – linear case

For Banach space E let $E = E^s \oplus E^u$.

Let us have a linear operator $A \in B(E)$ and assume that E^s i E^u are A-invariant. Then (in matrix form):

$$A = \begin{bmatrix} A_{ss} & 0 \\ 0 & A_{uu} \end{bmatrix}.$$

Definition

We say that operator A is hyperbolic if

- $\|A_{ss}\| \leq \lambda_s$, $\|A_{uu}^{-1}\| \leq \lambda_u^{-1}$;
- $\lambda_s < 1 < \lambda_u$;

for some equivalent norm $\|\cdot\|$ on E.

Hyperbolicity – linear case

For Banach space E let $E = E^s \oplus E^u$.

Let us have a linear operator $A \in B(E)$ and assume that E^s i E^u are A-invariant. Then (in matrix form):

$$A = \begin{bmatrix} A_{ss} & 0 \\ 0 & A_{uu} \end{bmatrix}.$$

Definition

We say that operator A is hyperbolic if

- $\|A_{ss}\| \leq \lambda_s$, $\|A_{uu}^{-1}\| \leq \lambda_u^{-1}$;
- $\lambda_s < 1 < \lambda_u$;

for some equivalent norm $\|\cdot\|$ on E.

Semi-hyperbolicity – linear case

For Banach space E let $E = E^s \oplus E^u$.

Let us have a linear operator $A \in B(E)$ and do not assume that E^s i E^u are A-invariant. Then (in matrix form):

$$A = \begin{bmatrix} A_{ss} & A_{su} \\ A_{us} & A_{uu} \end{bmatrix}.$$

Definition

We say that operator A is semi-hyperbolic if

- $\|A_{ss}\| \leq \lambda_s$, $\|A_{uu}^{-1}\| \leq \lambda_u^{-1}$, $\|A_{su}\| \leq \mu_s$, $\|A_{us}\| \leq \mu_u$;
- $\lambda_s < 1 < \lambda_u$, $(1 - \lambda_s)(\lambda_u - 1) > \mu_s \mu_u$;

for some equivalent norm $\|\cdot\|$ on E.

Problem discretization

Definition

Let G be a finite family of subsets of $\mathbb{R}^n$ and let $F\colon G \rightrightarrows G$, $DF\colon G \rightrightarrows B(\mathbb{R}^n)$ be multivalued maps. By $B(\mathbb{R}^n)$ we mean the space of bounded linear operators over $\mathbb{R}^n$ with usual norm. For each $\sigma \in G$ let $DF(\sigma)$ be a set-operator in $\mathbb{R}^n$ (bounded subset of $B(\mathbb{R}^n)$).

We say that f inherits dynamics of the pair (F, DF) on the set K (what we denote by $f \lhd_K (F, DF)$) iff

- $K \subset \bigcup_{\sigma \in \mathrm{dom}(F)} \sigma$;
- $K \cap f(\sigma) \cap \tau \neq \emptyset$ *for some* $\sigma, \tau \in G \Rightarrow \tau \in F(\sigma)$;
- $x \in K \cap \sigma$ *for some* $\sigma \in G \Rightarrow D_x f \in DF(\sigma)$.

Discrete semi-hyperbolicity

Definition

We say that the pair (F, DF) is $(\lambda_s, \lambda_u, \mu_s, \mu_u)$-semi-hyperbolic if for each $\sigma \in G$ there exist a norm $\|\cdot\|_\sigma$ on $\mathbb{R}^n$ and a decomposition $\mathbb{R}^n = E^s_\sigma \oplus E^u_\sigma$ such that for each $\sigma \in G, \tau \in F(\sigma)$ set-operator

$$DF(\sigma) : (E^s_\sigma \oplus E^u_\sigma, \|\cdot\|_\sigma) \to (E^s_\tau \oplus E^u_\tau, \|\cdot\|_\tau)$$

is $(\lambda_s, \lambda_u, \mu_s, \mu_u)$-semi-hyperbolic.

Theorem (Mazur, Tabor, Kościelniak)

Let (F, DF) be $(\lambda_s, \lambda_u, \mu_s, \mu_u)$-semi-hyperbolic pair such that $f \lhd_K (F, DF)$. Then K is $(\lambda_s, \lambda_u, \mu_s, \mu_u)$-semi-hyperbolic. As a consequence K is a hyperbolic set.

Semi-hyperbolicity in $\mathbb{R}^2$

Theorem

Let $\lambda_s \in [0,1), \lambda_u > 1, \mu_s \geq 0, \mu_u \geq 0$ *satisfy*

$$(\lambda_u - 1)(1 - \lambda_s) > \mu_s \mu_u.$$

If for every $\sigma \in G$ *we have a base* $B_\sigma = [e_s^\sigma, e_u^\sigma]$ *for* $\mathbb{R}^2$ *such that for every* $\sigma, \tau \in F(\sigma)$ *we have*

$$B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \subset \begin{bmatrix} [-\lambda_s, \lambda_s] & [-\mu_s, \mu_s] \\ [-\mu_u, \mu_u] & \mathbb{R} \backslash [-\lambda_u, \lambda_u] \end{bmatrix}$$

then (F, DF) *is* $(\lambda_s, \lambda_u, \mu_s, \mu_u)$*-semi-hyperbolic.*

Discretization

We start with the cube representation of the invariant part of the square $R = \left[-\frac{C}{2}, \frac{C}{2}\right] \times \left[-\frac{C}{2}, \frac{C}{2}\right]$, where $C = 1 + |b| + \sqrt{(1 + |b|)^2 + 4a}$.

First we cover the set R with 1×1-sized cubes.

Then we construct the discrete representation F of the map f.

Discretization

We start with the cube representation of the invariant part of the square $R = \left[-\frac{C}{2}, \frac{C}{2}\right] \times \left[-\frac{C}{2}, \frac{C}{2}\right]$, where $C = 1 + |b| + \sqrt{(1 + |b|)^2 + 4a}$.

First we cover the set R with 1×1-sized cubes.

Then we construct the discrete representation F of the map f.

Discretization

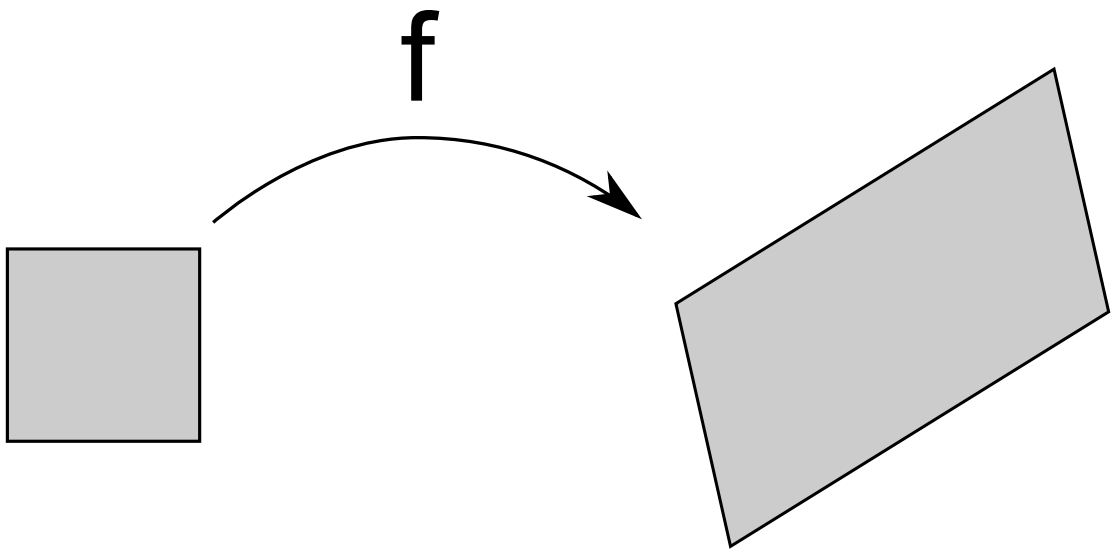
We start with the cube representation of the invariant part of the square $R = \left[-\frac{C}{2}, \frac{C}{2}\right] \times \left[-\frac{C}{2}, \frac{C}{2}\right]$, where $C = 1 + |b| + \sqrt{(1 + |b|)^2 + 4a}$.

First we cover the set R with 1×1-sized cubes.

Then we construct the discrete representation F of the map f.

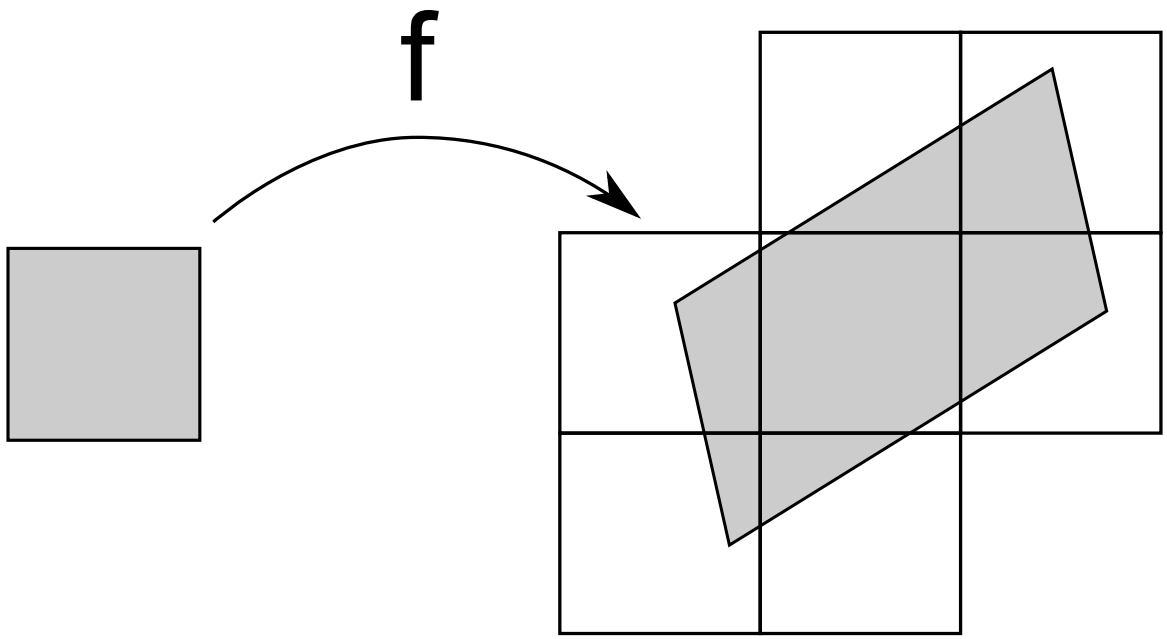
Discretization

For each cube we find its image under f using interval arithmetics and cover it with cubes of the same size.



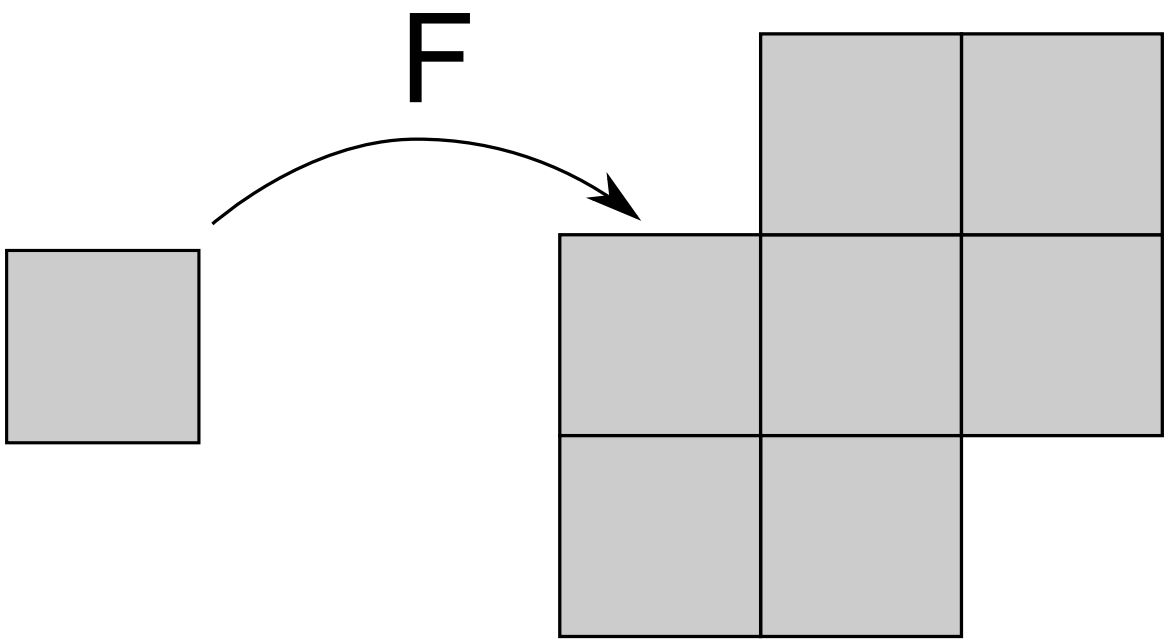
Discretization

For each cube we find its image under f using interval arithmetics and cover it with cubes of the same size.



Discretization

For each cube we find its image under f using interval arithmetics and cover it with cubes of the same size.



Invariant set

We iterate the following procedure:

For each cube σ from the domain of F check the conditions:

- $\sigma \in dom(F^{-1})$
- $F(\sigma) \cap dom(F) \neq \emptyset$
- $F(\sigma) \cap dom(F^{-1}) \neq \emptyset$
- $F^{-1}(\sigma) \cap dom(F) \neq \emptyset$
- $F^{-1}(\sigma) \cap dom(F^{-1}) \neq \emptyset$

If at least one check fails then remove σ from the domain.

Repeat until no such σ can be removed.

Invariant set

We iterate the following procedure:

For each cube σ from the domain of F check the conditions:

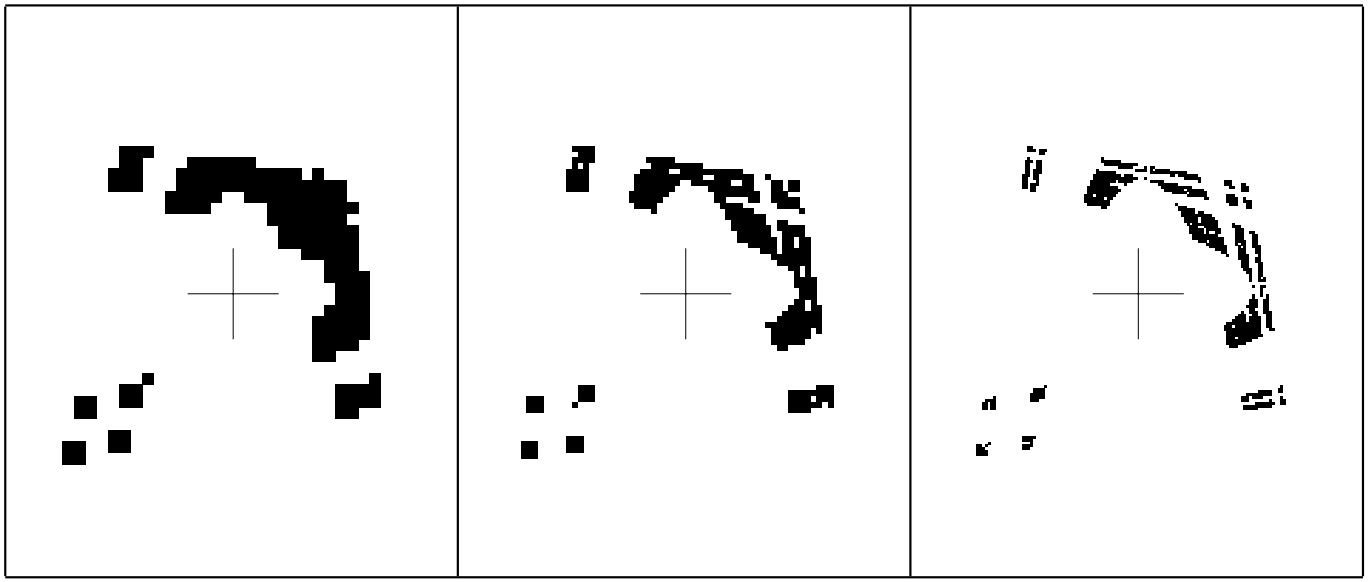
- $\sigma \in dom(F^{-1})$
- $F(\sigma) \cap dom(F) \neq \emptyset$
- $F(\sigma) \cap dom(F^{-1}) \neq \emptyset$
- $F^{-1}(\sigma) \cap dom(F) \neq \emptyset$
- $F^{-1}(\sigma) \cap dom(F^{-1}) \neq \emptyset$

If at least one check fails then remove σ from the domain.

Repeat until no such σ can be removed.

Invariant set

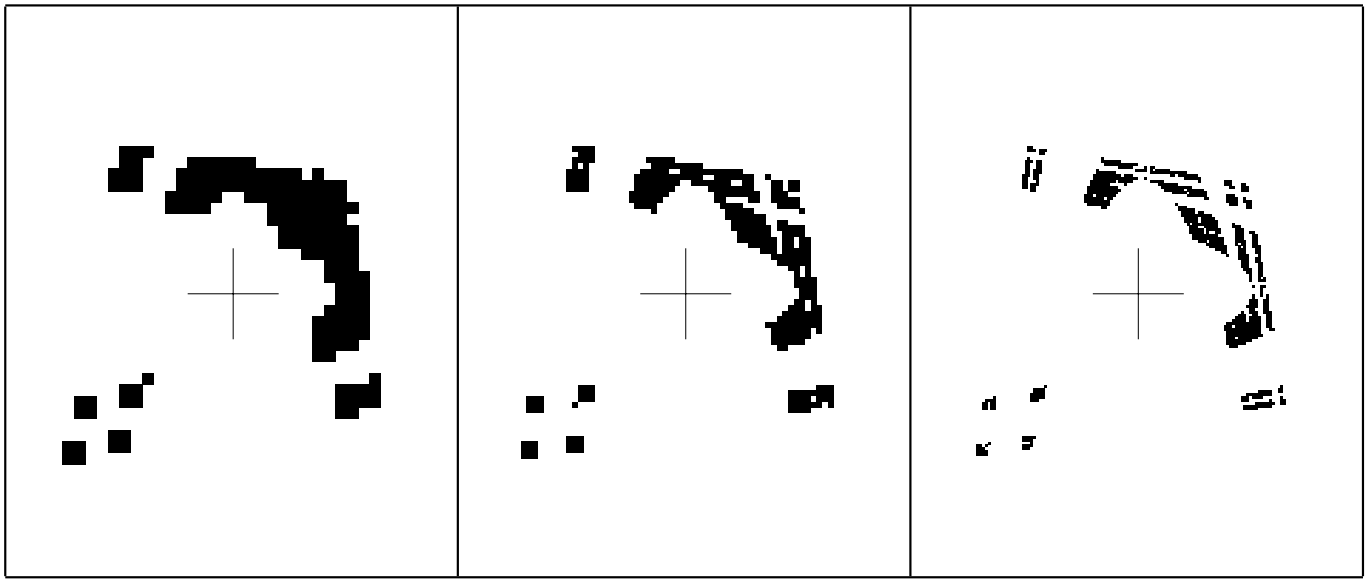
We get the covering of the invariant set with cubes of size 1×1. Next we divide each cube into 4 smaller ones and again we construct discretization F and find outer approximation of the invariant set. We stop when the assumed level of the precision is reached.



We get the representation of DF using interval arithmetics (we have the explicit formula for the derivative).

Invariant set

We get the covering of the invariant set with cubes of size 1×1. Next we divide each cube into 4 smaller ones and again we construct discretization F and find outer approximation of the invariant set. We stop when the assumed level of the precision is reached.



We get the representation of DF using interval arithmetics (we have the explicit formula for the derivative).

Decomposition of $\mathbb{R}^2$

Having the outer approximation of the invariant set we try to find for each cube σ the decomposition of $\mathbb{R}^2$ into stable and unstable subspaces (E^u_σ i E^s_σ).

We use one-valued simplifications of F and F^{-1} ($F_1 : G \longrightarrow G$ i $F_1^{-1} : G \longrightarrow G$ respectively). We define these functions using only the graph representation F to make this part independent of the fact that we know exactly the formula for f.

We also define $DF_1 : G \longrightarrow B(\mathbb{R}^n)$ as the arbitrarily chosen operator from $DF(\sigma)$. It can be central point from $DF(\sigma)$ or $D_{x_\sigma} f$ where x_σ is the central point from σ.
Similarly we define DF_1^{-1}.

Decomposition of $\mathbb{R}^2$

Having the outer approximation of the invariant set we try to find for each cube σ the decomposition of $\mathbb{R}^2$ into stable and unstable subspaces (E^u_σ i E^s_σ).

We use one-valued simplifications of F and F^{-1} ($F_1 \colon G \longrightarrow G$ i $F_1^{-1} \colon G \longrightarrow G$ respectively). We define these functions using only the graph representation F to make this part independent of the fact that we know exactly the formula for f.

We also define $DF_1 \colon G \longrightarrow B(\mathbb{R}^n)$ as the arbitrarily chosen operator from $DF(\sigma)$. It can be central point from $DF(\sigma)$ or $D_{x_\sigma} f$ where x_σ is the central point from σ.
Similarly we define DF_1^{-1}.

Decomposition of $\mathbb{R}^2$

Having the outer approximation of the invariant set we try to find for each cube σ the decomposition of $\mathbb{R}^2$ into stable and unstable subspaces (E^u_σ i E^s_σ).

We use one-valued simplifications of F and F^{-1} ($F_1 \colon G \longrightarrow G$ i $F_1^{-1} \colon G \longrightarrow G$ respectively). We define these functions using only the graph representation F to make this part independent of the fact that we know exactly the formula for f.

We also define $DF_1 \colon G \longrightarrow B(\mathbb{R}^n)$ as the arbitrarily chosen operator from $DF(\sigma)$. It can be central point from $DF(\sigma)$ or $D_{x_\sigma} f$ where x_σ is the central point from σ.
Similarly we define DF_1^{-1}.

unstable (stable) direction

We fix a number $k \in \mathbb{N}$ (for example 10). For each cube $\sigma \in G$ we define two matrices

$$A = DF_1(F_1^k(\sigma)) \circ \cdots \circ DF_1(F_1(\sigma)) \circ DF_1(\sigma),$$

$$B = DF_1(F_1^{-1}(\sigma)) \circ \cdots \circ DF_1(F_1^{-(k+1)}(\sigma)).$$

With the power method we find the vector v_u with the greatest absolute eigenvalue of the operator $A \circ B$ and as the unstable direction we take $e_\sigma^u := B(v_u)$.

For the stable direction we do the same where

$$A = DF_1^{-1}(F_1^{-k}(\sigma)) \circ \cdots \circ DF_1^{-1}(\sigma),$$

$$B = DF_1^{-1}(F_1(\sigma)) \circ \cdots \circ DF_1^{-1}(F_1^{(k+1)}(\sigma)).$$

unstable (stable) direction

We fix a number $k \in \mathbb{N}$ (for example 10). For each cube $\sigma \in G$ we define two matrices

$$A = DF_1(F_1^k(\sigma)) \circ \cdots \circ DF_1(F_1(\sigma)) \circ DF_1(\sigma),$$

$$B = DF_1(F_1^{-1}(\sigma)) \circ \cdots \circ DF_1(F_1^{-(k+1)}(\sigma)).$$

With the power method we find the vector v_u with the greatest absolute eigenvalue of the operator $A \circ B$ and as the unstable direction we take $e_\sigma^u := B(v_u)$.

For the stable direction we do the same where

$$A = DF_1^{-1}(F_1^{-k}(\sigma)) \circ \cdots \circ DF_1^{-1}(\sigma),$$

$$B = DF_1^{-1}(F_1(\sigma)) \circ \cdots \circ DF_1^{-1}(F_1^{(k+1)}(\sigma)).$$

unstable (stable) direction

We fix a number $k \in \mathbb{N}$ (for example 10). For each cube $\sigma \in G$ we define two matrices

$$A = DF_1(F_1^k(\sigma)) \circ \cdots \circ DF_1(F_1(\sigma)) \circ DF_1(\sigma),$$

$$B = DF_1(F_1^{-1}(\sigma)) \circ \cdots \circ DF_1(F_1^{-(k+1)}(\sigma)).$$

With the power method we find the vector v_u with the greatest absolute eigenvalue of the operator $A \circ B$ and as the unstable direction we take $e_\sigma^u := B(v_u)$.

For the stable direction we do the same where

$$A = DF_1^{-1}(F_1^{-k}(\sigma)) \circ \cdots \circ DF_1^{-1}(\sigma),$$

$$B = DF_1^{-1}(F_1(\sigma)) \circ \cdots \circ DF_1^{-1}(F_1^{(k+1)}(\sigma)).$$

Renormalisation

Finally we may need to rescale the base vectors e^u_σ and e^s_σ. They were chosen to have the norm arbitrarily equal to 1. What we need is that for $\tau \in F(\sigma)$ the operators of the form

$$Pr_{E^u_\tau} \circ DF_1(\sigma)|_{E^u_\sigma} \colon E^u_\sigma \longrightarrow E^u_\tau$$

should be in a sense as much expansive as possible.

Renormalisation

We set another parameter $l \in \mathbb{N}$ and for $\sigma \in G$ we regard a part of the approximate orbit
$\sigma_{-l} := F_1^{-l}(\sigma), \ldots, \sigma, \ldots, \sigma_l := F_1^l(\sigma)$.

Let $e_0^u := e_\sigma^u$, $e_0^s := e_\sigma^s$, $m_0^u := m_0^s := 1$. For $i = 1, \ldots, l$ define

$$e_i^u := Pr_{E^u_{\sigma_i}}(DF_1(\sigma_{i-1})e_{i-1}^u),$$

$$m_i^u := \|e_i^u\|.$$

We do the same for $i = -1, \ldots, -l$.

Renormalisation

We set another parameter $l \in \mathbb{N}$ and for $\sigma \in G$ we regard a part of the approximate orbit
$\sigma_{-l} := F_1^{-l}(\sigma), \ldots, \sigma, \ldots, \sigma_l := F_1^l(\sigma)$.

Let $e_0^u := e_\sigma^u$, $e_0^s := e_\sigma^s$, $m_0^u := m_0^s := 1$. For $i = 1, \ldots, l$ define

$$e_i^u := Pr_{E^u_{\sigma_i}}(DF_1(\sigma_{i-1})e_{i-1}^u),$$

$$m_i^u := \|e_i^u\|.$$

We do the same for $i = -1, \ldots, -l$.

Renormalisation

We set another parameter $l \in \mathbb{N}$ and for $\sigma \in G$ we regard a part of the approximate orbit
$\sigma_{-l} := F_1^{-l}(\sigma), \ldots, \sigma, \ldots, \sigma_l := F_1^l(\sigma)$.

Let $e_0^u := e_\sigma^u$, $e_0^s := e_\sigma^s$, $m_0^u := m_0^s := 1$. For $i = 1, \ldots, l$ define

$$e_i^u := Pr_{E^u_{\sigma_i}}(DF_1(\sigma_{i-1})e_{i-1}^u),$$

$$m_i^u := \|e_i^u\|.$$

We do the same for $i = -1, \ldots, -l$.

Renormalisation

Having the sequence $m^u_{-l}, \ldots, m^u_l$ we calculate the mean expansion along the approximate orbit as $\lambda := \left(\frac{m^u_l}{m^u_{-l}}\right)^{1/2l}$ and define

$$M^u := \max\left\{ \max_{i=0,\ldots,l}\left\{ m^u_i \left(\frac{1}{\lambda\delta}\right)^i \right\}, \max_{i=-1,\ldots,-l}\left\{ m^u_i \left(\frac{\delta}{\lambda}\right)^i \right\} \right\},$$

where $\delta = 1 + \varepsilon$.

Finally we fix e^u_σ to be equal e^u_σ / M^u.

The case for stable direction is similar.

Renormalisation

Having the sequence $m^u_{-l}, \ldots, m^u_l$ we calculate the mean expansion along the approximate orbit as $\lambda := \left(\frac{m^u_l}{m^u_{-l}}\right)^{1/2l}$ and define

$$M^u := \max\left\{\max_{i=0,\ldots,l}\left\{m^u_i\left(\frac{1}{\lambda\delta}\right)^i\right\}, \max_{i=-1,\ldots,-l}\left\{m^u_i\left(\frac{\delta}{\lambda}\right)^i\right\}\right\},$$

where $\delta = 1 + \varepsilon$.

Finally we fix e^u_σ to be equal e^u_σ / M^u.

The case for stable direction is similar.

Renormalisation

Having the sequence $m^u_{-l}, \ldots, m^u_l$ we calculate the mean expansion along the approximate orbit as $\lambda := \left(\frac{m^u_l}{m^u_{-l}}\right)^{1/2l}$ and define

$$M^u := \max\left\{\max_{i=0,\ldots,l}\left\{m^u_i\left(\frac{1}{\lambda\delta}\right)^i\right\}, \max_{i=-1,\ldots,-l}\left\{m^u_i\left(\frac{\delta}{\lambda}\right)^i\right\}\right\},$$

where $\delta = 1 + \varepsilon$.

Finally we fix e^u_σ to be equal e^u_σ / M^u.

The case for stable direction is similar.

Verifying semi-hyperbolicity

Having defined bases $B_\sigma = [e^s_\sigma, e^u_\sigma]$ we want to check the assumptions of the theorem. That is if there exists constants $\lambda_s, \lambda_u, \mu_s, \mu_u$ such that for each $\sigma \in G$, $\tau \in F(\sigma)$ the following conditions are satisfied

$$B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \subset \begin{bmatrix} [-\lambda_s, \lambda_s] & [-\mu_s, \mu_s] \\ [-\mu_u, \mu_u] & \mathbb{R}\backslash[-\lambda_u, \lambda_u] \end{bmatrix}.$$

To check that we calculate

$$\lambda_s := \max_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(1,1)}| : A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\},$$

$$\mu_s := \max_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(1,2)}| : A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\},$$

$$\mu_u := \max_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(2,1)}| : A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\},$$

$$\lambda_u := \min_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(2,2)}| : A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\}.$$

Verifying semi-hyperbolicity

Having defined bases $B_\sigma = [e^s_\sigma, e^u_\sigma]$ we want to check the assumptions of the theorem. That is if there exists constants $\lambda_s, \lambda_u, \mu_s, \mu_u$ such that for each $\sigma \in G$, $\tau \in F(\sigma)$ the following conditions are satisfied

$$B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \subset \begin{bmatrix} [-\lambda_s, \lambda_s] & [-\mu_s, \mu_s] \\ [-\mu_u, \mu_u] & \mathbb{R}\backslash[-\lambda_u, \lambda_u] \end{bmatrix}.$$

To check that we calculate

$$\lambda_s := \max_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(1,1)}| \colon A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\},$$

$$\mu_s := \max_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(1,2)}| \colon A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\},$$

$$\mu_u := \max_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(2,1)}| \colon A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\},$$

$$\lambda_u := \min_{\sigma,\tau \in F(\sigma)} \left\{ |A^{\sigma,\tau}_{(2,2)}| \colon A^{\sigma,\tau} \in B_\tau \circ DF(\sigma) \circ (B_\sigma)^{-1} \right\}.$$

Verifying semi-hyperbolicity

If such defined $\lambda_s, \lambda_u, \mu_s, \mu_u$ satisfy $\lambda_s < 1, \lambda_u > 1$ and

$$(\lambda_u - 1)(1 - \lambda_s) > \mu_s \mu_u$$

then (F, DF) is semi-hyperbolic and as a consequence the invariant set, which covering we constructed, is hyperbolic.

For instance for $(a, b) \in \{(5.4, -1), (5.5, -0.8), (3, 0.25)\}$ the invariant part of our starting square R is hyperbolic.

Object-oriented programming

Program uses virtual classes, interfaces etc.

It allows easy and fast modification of previously separated fragments.

For example the approximation of the invariant set for the problem we deal with is a distinct issue. It can be solved in many different ways. With object-oriented programming one can change the way to solve it quite easily.

Object-oriented programming

Program uses virtual classes, interfaces etc.

It allows easy and fast modification of previously separated fragments.

For example the approximation of the invariant set for the problem we deal with is a distinct issue. It can be solved in many different ways. With object-oriented programming one can change the way to solve it quite easily.

Object-oriented programming

Program uses virtual classes, interfaces etc.

It allows easy and fast modification of previously separated fragments.

For example the approximation of the invariant set for the problem we deal with is a distinct issue. It can be solved in many different ways. With object-oriented programming one can change the way to solve it quite easily.

Standard Template Library

Program is based on the Standard Template Library *STL* (www.sgi.com/tech/stl/) which is a part of C++ Standard Library.

- implementation is easy;
- templates allows easy switch to interval arithmetics from standard one;
- platform independence;
- program editing possible (theoreticaly) for non-authors.

Standard Template Library

Program is based on the Standard Template Library *STL* (www.sgi.com/tech/stl/) which is a part of C++ Standard Library.

- implementation is easy;
- templates allows easy switch to interval arithmetics from standard one;
- platform independence;
- program editing possible (theoreticaly) for non-authors.

Standard Template Library

Program is based on the Standard Template Library *STL* (www.sgi.com/tech/stl/) which is a part of C++ Standard Library.

- implementation is easy;
- templates allows easy switch to interval arithmetics from standard one;
- platform independence;
- program editing possible (theoreticaly) for non-authors.

Standard Template Library

Program is based on the Standard Template Library *STL* (www.sgi.com/tech/stl/) which is a part of C++ Standard Library.

- implementation is easy;
- templates allows easy switch to interval arithmetics from standard one;
- platform independence;
- program editing possible (theoreticaly) for non-authors.

Standard Template Library

Program is based on the Standard Template Library *STL* (www.sgi.com/tech/stl/) which is a part of C++ Standard Library.

- implementation is easy;
- templates allows easy switch to interval arithmetics from standard one;
- platform independence;
- program editing possible (theoreticaly) for non-authors.

Standard Template Library

On the basis of STL classes like set, map etc. we defined our own objects

- CUBE - cube representation in $\mathbb{R}^n$;
- SET - representation of the set of cubes;
- MAP - multivalued function mapping cubes into sets of cubes;
- DERIVATIVE_MAP - representation of the set-operator.

Standard Template Library

On the basis of STL classes like set, map etc. we defined our own objects

- CUBE - cube representation in $\mathbb{R}^n$;
- SET - representation of the set of cubes;
- MAP - multivalued function mapping cubes into sets of cubes;
- DERIVATIVE_MAP - representation of the set-operator.

Standard Template Library

Additional objects:

- FUNCTION - representation of the function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$;
- DERIVATIVE - representation of the derivative;
- others.

Boost

Program uses *boost* C++ library (http://www.boost.org/) from which we utilized interval arithmetics.

It is not a part of C++ Standard Library hence manual installation is needed.

Boost

Program uses *boost* C++ library (http://www.boost.org/) from which we utilized interval arithmetics.

It is not a part of C++ Standard Library hence manual installation is needed.

Thank you very much