Rigorous computation of Poincaré maps

T. Kapela, D. Wilczak, P. Zgliczyński

Jagiellonian University, Kraków, Poland

SWIM '2015

June 11, 2015, Praha, Česká Republika

Outline of presentation:

- theory of Poincaré maps
- an algorithm for computation of Poincaré maps
- choice of sections and coordinate systems (examples)

Part 1

Theory

Local sections

Definition

$\Pi \subset \mathbb{R}^n$ is δ**-section** for $(t, x) \to \varphi(t, x)$

$$\Updownarrow$$

$(-\delta, \delta) \times \Pi \ni (t, x) \to \varphi(t, x)$ is diffeomorphism onto image

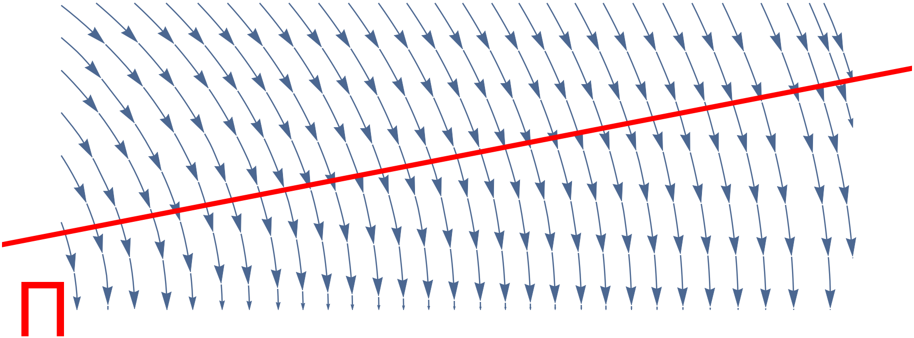
Local sections

Definition

$\Pi \subset \mathbb{R}^n$ is δ**-section** for $(t, x) \to \varphi(t, x)$

$$\Updownarrow$$

$(-\delta, \delta) \times \Pi \ni (t, x) \to \varphi(t, x)$ is diffeomorphism onto image



Poincaré sections

Definition

Π is *Poincaré section* for $(t, x) \to \varphi(t, x)$

$$\Updownarrow$$

Π is locally δ-section for some $\delta > 0$

Remark

For Π smooth and $x' = f(x)$ it is enough to have $\langle f(x); n_S(x)\rangle \neq 0$

Poincaré sections

Definition

Π is *Poincaré section* for $(t, x) \to \varphi(t, x)$

$$\Updownarrow$$

Π is locally δ-section for some $\delta > 0$

Remark

For Π smooth and $x' = f(x)$ it is enough to have $\langle f(x); n_S(x)\rangle \neq 0$

Poincaré maps

Π_1, Π_2 - Poincaré sections for φ

Definition

$P : \Pi_1 \to \Pi_2$ - Poincaré map

- $x \in \mathrm{dom}(P)$ iff $\varphi(t, x) \in \Pi_2$ for some $t > 0$
- $P(x)$ - first cut of $\varphi(t, x)$ with Π_2 for $t > 0$

Poincaré maps

Π_1, Π_2 - Poincaré sections for φ

Definition

$P\colon \Pi_1 \to \Pi_2$ - Poincaré map

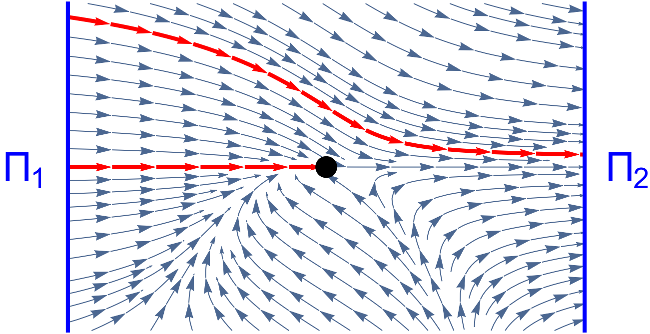
- $x \in \mathrm{dom}(P)$ iff $\varphi(t,x) \in \Pi_2$ for some $t > 0$
- $P(x)$ - first cut of $\varphi(t,x)$ with Π_2 for $t > 0$

Π_1, Π_2 - Poincaré sections for φ

Definition

$P \colon \Pi_1 \to \Pi_2$ - Poincaré map

- $x \in \mathrm{dom}(P)$ iff $\varphi(t, x) \in \Pi_2$ for some $t > 0$
- $P(x)$ - first cut of $\varphi(t, x)$ with Π_2 for $t > 0$



Practical description of Poincaré sections

$$\Pi = \Pi_{\alpha,\mathcal{C}} = \{x : \alpha(\mathbf{x}) = \mathbf{0} \wedge \langle \nabla\alpha(x); f(x)\rangle \neq 0 \wedge \mathcal{C}(\mathbf{x})\}$$

where

- $\alpha\colon \mathbb{R}^n \to \mathbb{R}$ - smooth
- zero is a **regular value** of α
- $\mathcal{C}$ **is a predicate** (additional constrains on the section)
  - crossing direction
  - restriction on the domain
  - etc.

Settings

- Π_1, Π_2 - sections given by $\alpha_i : \mathbb{R}^n \to \mathbb{R}$
- $P \colon \Pi_1 \to \Pi_2$ - Poincaré map

Question: is P continuous? smooth?

Settings

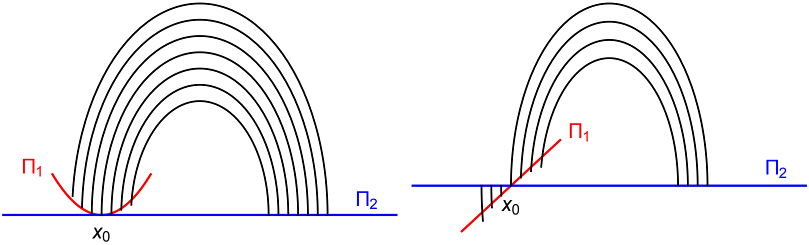
- Π_1, Π_2 - sections given by $\alpha_i : \mathbb{R}^n \to \mathbb{R}$
- $P \colon \Pi_1 \to \Pi_2$ - Poincaré map

Question: is P continuous? smooth?

Settings

- Π_1, Π_2 - sections given by $\alpha_i : \mathbb{R}^n \to \mathbb{R}$
- $P \colon \Pi_1 \to \Pi_2$ - Poincaré map

Question: is P continuous? smooth?



Theorem

If

- $\Pi_i = \mathrm{cl}(\mathrm{int}\Pi_i)$
- *either* $\Pi_1 \subset \Pi_2$ *or* $\Pi_1 \cap \Pi_2 = \emptyset$

Then $P\colon \Pi_1 \to \Pi_2$ *is smooth at every point* $x \in \mathrm{dom}P \cap \mathrm{int}\Pi_1$ *such that* $P(x) \in \mathrm{int}\Pi_2$*.*

Goals

- Give an algorithm for enclosing $P(X)$, $X \subset \Pi_1$
- Discuss how results depend on the choice of Poincaré sections and coordinate systems

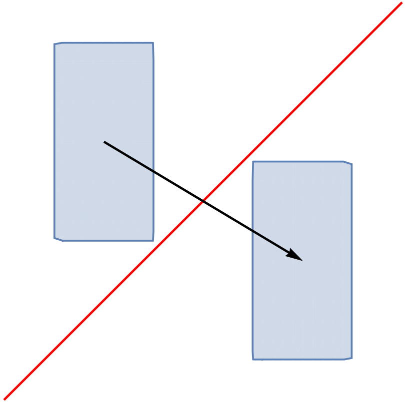
Part 2

Algorithm

Enclosing Poincaré maps

Constrains:

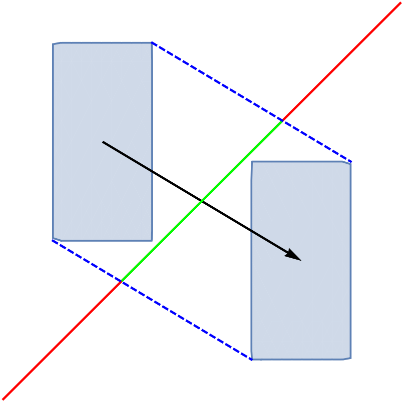
- avoid subdivisions
- reduce wrapping effect



Enclosing Poincaré maps

Constrains:

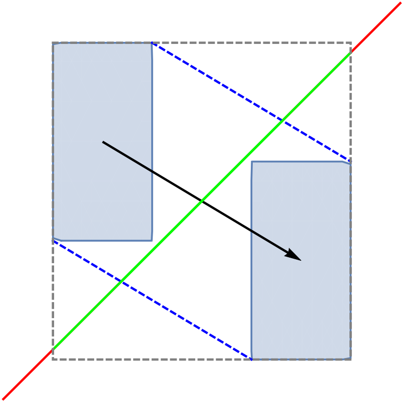
- avoid subdivisions
- reduce wrapping effect



Enclosing Poincaré maps

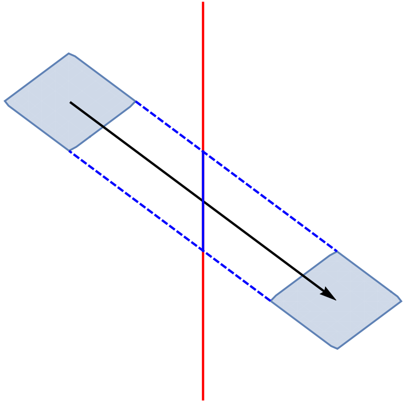
Constrains:

- avoid subdivisions
- reduce wrapping effect



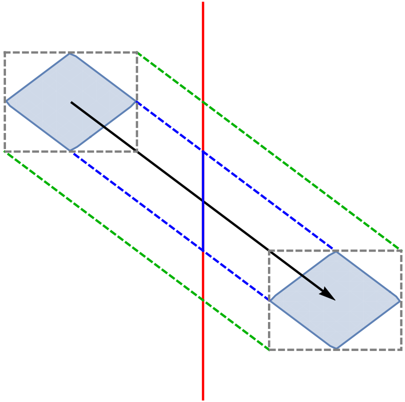
Enclosing Poincaré maps

Very important:
take into account internal representation of solutions in ODE solver



Enclosing Poincaré maps

Very important:
take into account internal representation of solutions in ODE solver



Abstract data structure: `RepresentableSet`

Example:

$$X = x + Cr_0 + Br$$

Abstract algorithm:

Algorithm: AFFINETRANSFORM

Input: $X \subset \mathbb{R}^n$ - `RepresentableSet`
Input: $Q : \mathbb{R}^n \to \mathbb{R}^m$ - linear map
Input: $x_0 \in \mathbb{R}^n$ - vector
Output: Bound for $Q(X - x_0)$

Example:

$$Q(x - x_0 + Cr_0 + Br) \subset (Q(x - x_0) + (QC)r_0 + (QB)r)$$

Abstract data structure: `RepresentableSet`

Example:

$$X = x + Cr_0 + Br$$

Abstract algorithm:

Algorithm: AFFINETRANSFORM

Input: $X \subset \mathbb{R}^n$ - `RepresentableSet`
Input: $Q : \mathbb{R}^n \to \mathbb{R}^m$ - linear map
Input: $x_0 \in \mathbb{R}^n$ - vector
Output: Bound for $Q(X - x_0)$

Example:

$$Q(x - x_0 + Cr_0 + Br) \cap (Q(x - x_0) + (QC)r_0 + (QB)r)$$

Abstract data structure: `RepresentableSet`

Example:

$$X = x + Cr_0 + Br$$

Abstract algorithm:

Algorithm: AFFINETRANSFORM

Input: $X \subset \mathbb{R}^n$ - `RepresentableSet`
Input: $Q : \mathbb{R}^n \to \mathbb{R}^m$ - linear map
Input: $x_0 \in \mathbb{R}^n$ - vector
Output: Bound for $Q(X - x_0)$

Example:

$$Q(x - x_0 + Cr_0 + Br) \cap (Q(x - x_0) + (QC)r_0 + (QB)r)$$

Abstract data structure: `RepresentableSet`

Example:

$$X = x + Cr_0 + Br$$

Abstract algorithm:

Algorithm: AFFINETRANSFORM

Input: $X \subset \mathbb{R}^n$ - `RepresentableSet`
Input: $Q : \mathbb{R}^n \to \mathbb{R}^m$ - linear map
Input: $x_0 \in \mathbb{R}^n$ - vector
Output: Bound for $Q(X - x_0)$

Example:

$$Q(x - x_0 + Cr_0 + Br) \cap (Q(x - x_0) + (QC)r_0 + (QB)r)$$

Abstract algorithm:

Algorithm: EVAL

Input: $X \subset \mathbb{R}^n$ - `RepresentableSet`
Input: $g : \mathbb{R}^n \to \mathbb{R}^m$ - smooth
Output: Bound for $g(X)$

Example:

Algorithm: EVAL

Input: $x + Cr_0 + Br \subset \mathbb{R}^n$ - doubleton
Input: $g : \mathbb{R}^n \to \mathbb{R}^m$ - smooth function
Output: Bound for $g(x + Cr_0 + Br)$

// enclose set as interval vector
$X \leftarrow [x + Cr_0 + Br]_I$;
// enclose derivative as interval matrix
$M \leftarrow [Dg(X)]_I$;
return $[g(X)]_I \cap [g(x) + (MC)r_0 + (MB)r]_I$;

Abstract algorithm:

Algorithm: EVAL

Input: $X \subset \mathbb{R}^n$ - `RepresentableSet`
Input: $g : \mathbb{R}^n \to \mathbb{R}^m$ - smooth
Output: Bound for $g(X)$

Example:

Algorithm: EVAL

Input: $x + Cr_0 + Br \subset \mathbb{R}^n$ - doubleton
Input: $g : \mathbb{R}^n \to \mathbb{R}^m$ - smooth function
Output: Bound for $g(x + Cr_0 + Br)$

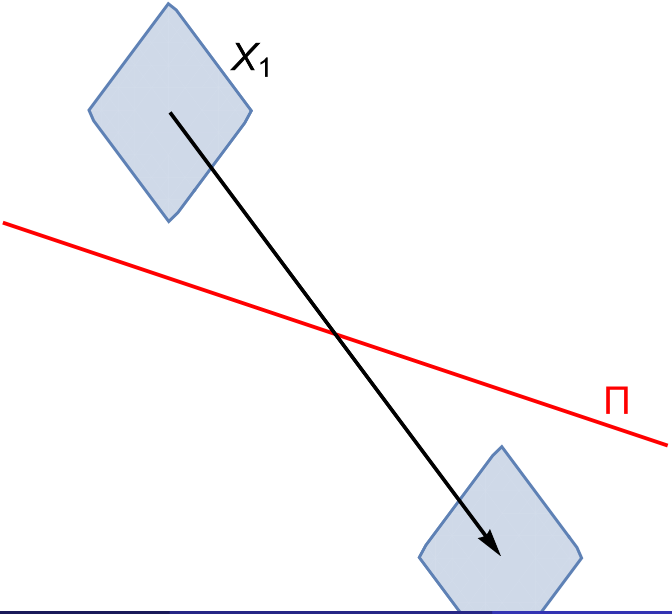
// enclose set as interval vector
$X \leftarrow [x + Cr_0 + Br]_I$;
// enclose derivative as interval matrix
$M \leftarrow [Dg(X)]_I$;
return $[g(X)]_I \cap [g(x) + (MC)r_0 + (MB)r]_I$;

Algorithm: COMPUTEPOINCAREMAP

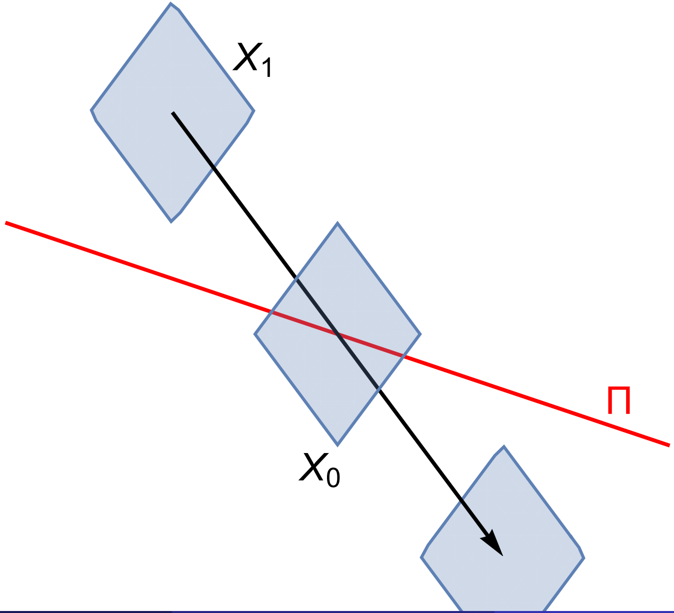
Input: $[t_1, t_2]$ - bound for return time
Input: X_1 - `RepresentableSet` that encloses $\varphi(t_1, X)$
Input: α - function that defines the section Π_2
Input: f - vector field that defines an ODE
Input: x_0 - a vector
Input: Q - a linear map
Output: Bound for $Q(P(X) - x_0)$

$t_0 \leftarrow (t_1 + t_2)/2$;
$\Delta t \leftarrow [t_1, t_2] - t_0$;
$X_0 \leftarrow$ `RepresentableSet` that encloses $\varphi(t_0 - t_1, X_1)$;
$Y_0 \leftarrow \mathtt{affineTransform}(X_0, Q, x_0)$;
$Y \leftarrow \mathtt{eval}(X_0, Q \circ f) \cdot \Delta t$;
$E \leftarrow \mathtt{eval}(X_1, \varphi([0, t_2 - t_1], \cdot))$;
$\Delta Y \leftarrow \frac{1}{2} Q \cdot Df(E) \cdot f(E) \cdot \Delta t^2$;
$Z \leftarrow (Y_0 + Y + \Delta Y) \cap Q(E - x_0)$;
return Z;

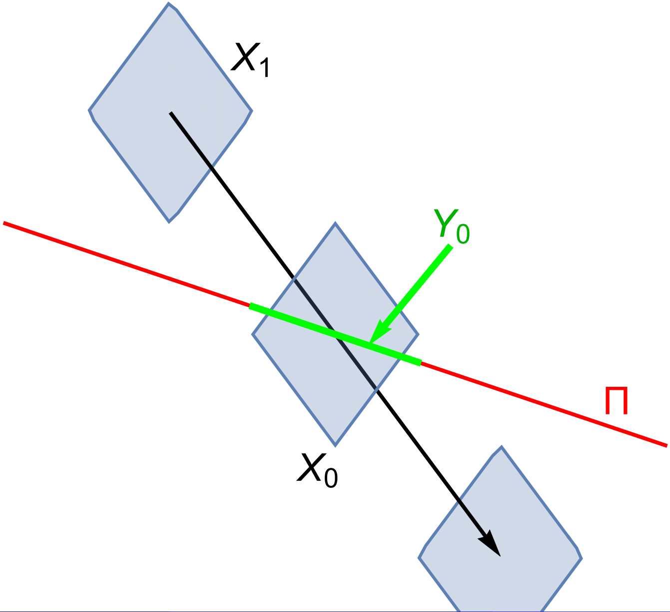
Enclosing Poincaré maps - geometry of the algorithm



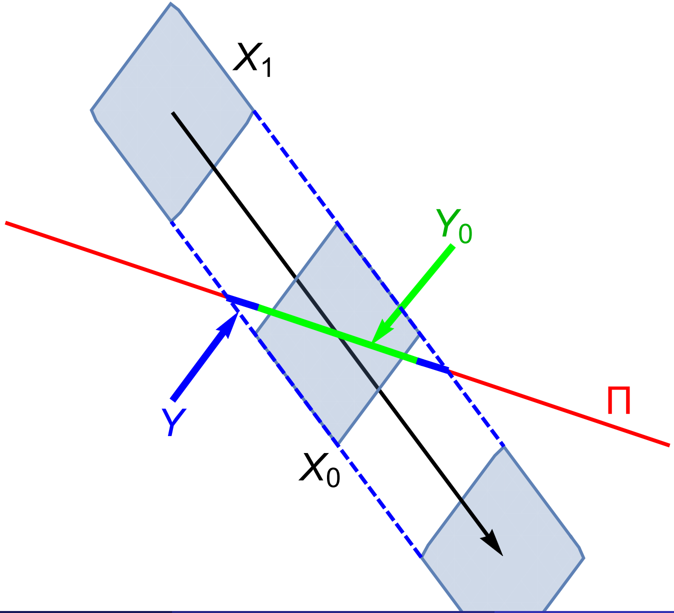
Enclosing Poincaré maps - geometry of the algorithm



Enclosing Poincaré maps - geometry of the algorithm



Enclosing Poincaré maps - geometry of the algorithm



Part 3a

Fixed section
Choosing coordinates

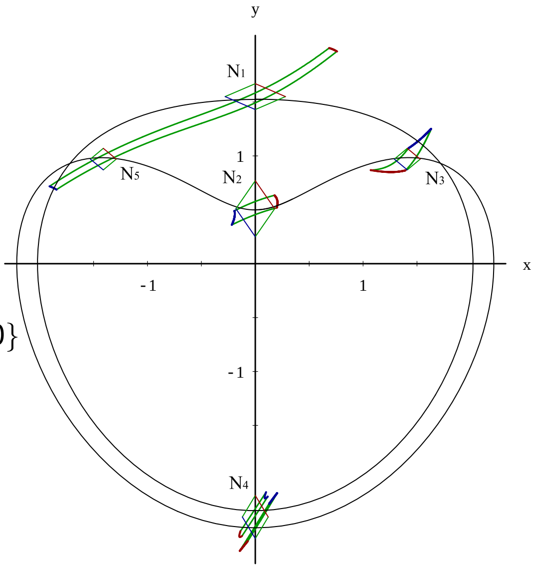
Michelson system

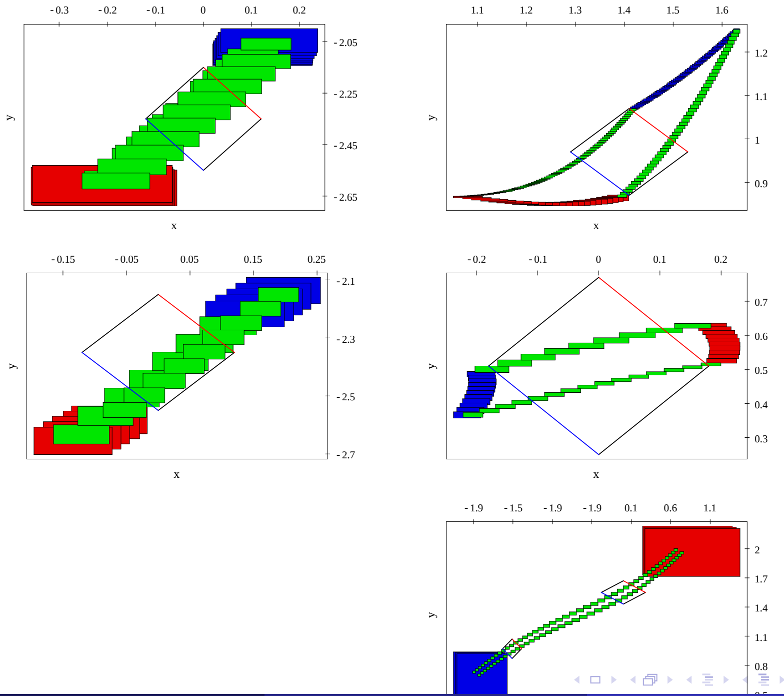
$$y''' = 1 - y' - \frac{1}{2}y^2$$

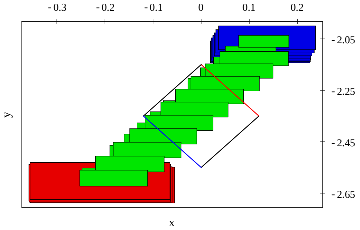
Poincaré section:

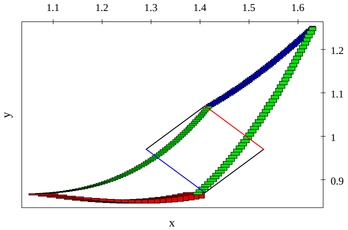
$$\Pi = \{(x, y, 0) \in \mathbb{R}^3 : z = 0\}$$

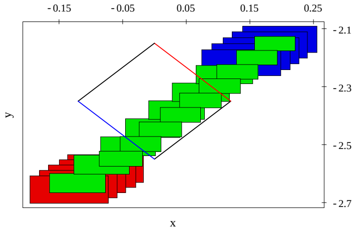
Chaos proved:
DW. J.Diff.Eq '2003.

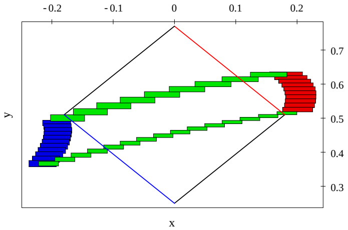



-0.3
-0.2
-0.1
0
0.1
0.2
-2.05
-2.25
-2.45
-2.65
x
y
1.1
1.2
1.3
1.4
1.5
1.6
1.2
1.1
1
0.9
x
y
-0.15
-0.05
0.05
0.15
0.25
-2.1
-2.3
-2.5
-2.7
x
y
-0.2
-0.1
0
0.1
0.2
0.7
0.6
0.5
0.4
0.3
x
y
-1.9
-1.5
-1.9
-1.9
0.1
0.6
1.1
2
1.7
1.4
1.1
0.8
y

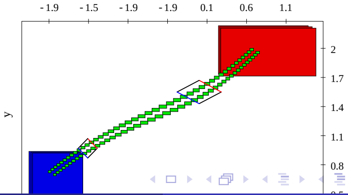








Show program

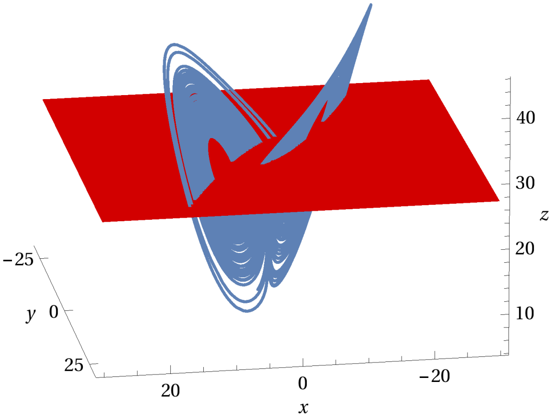


Lorenz system

$$\begin{cases} x' = 10(y-x) \\ y' = x(28-z) - y \\ z' = xy - \frac{8}{3}z \end{cases}$$

Poincaré section:

$$\Pi = \{(x,y,z) \in \mathbb{R}^3 : z = 27\}$$



Chaos proved:
Galias, Zgliczyński. Physica D '1998.

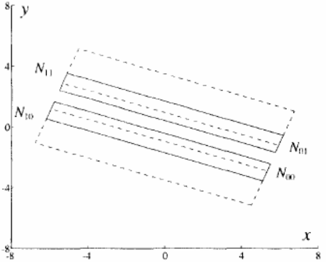


Fig. 2. Rectangles N_{00}, N_{01}, N_{10} and N_{11} on the transversal plane. N_{00} and N_{01} are printed with solid lines, while N_{10} and N_{11} with dashed ones.

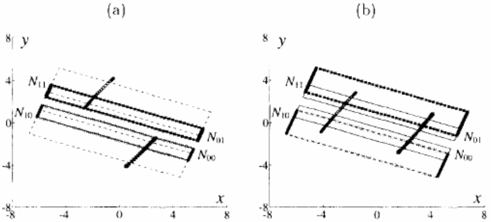
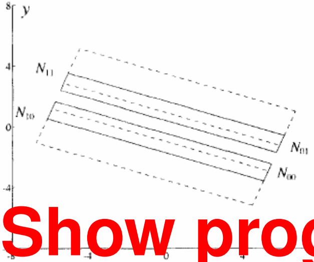


Fig. 3. Images of borders of N_{00}, N_{01}, N_{10} and N_{11} on the transversal plane – computer simulations: (a) Images of edges of N_{00} and N_{01}, one can clearly see that image of N_{00} covers N_{11} horizontally and symmetrically the image of N_{01} covers N_{10} and (b) images of edges of N_{10} and N_{11}, both images cover N_{00} and N_{01} horizontally.



Show program

Fig. 2. Rectangles N_{00}, N_{01}, N_{10} and N_{11} on the transversal plane. N_{00} and N_{01} are printed with solid lines, while N_{10} and N_{11} with dashed ones.

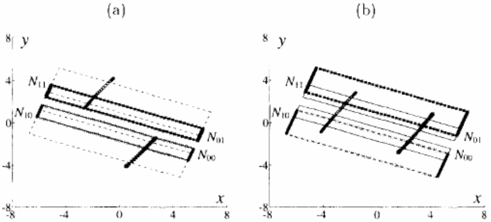


Fig. 3. Images of borders of N_{00}, N_{01}, N_{10} and N_{11} on the transversal plane – computer simulations: (a) Images of edges of N_{00} and N_{01}, one can clearly see that image of N_{00} covers N_{11} horizontally and symmetrically the image of N_{01} covers N_{10} and (b) images of edges of N_{10} and N_{11}, both images cover N_{00} and N_{01} horizontally.

Part 3b

Varying section: reduce “sliding”

Locally orthogonal sections

Minimize "sliding" part:

$$\begin{aligned} Y &= \texttt{eval}(X_0, Q \circ f) \cdot \Delta t \\ &\subset (Qf(\mathrm{mid}(X_0)) + QDf(X_0)(X_0 - \mathrm{mid}(X_0)))\Delta t \end{aligned}$$

The term

$$Qf(\mathrm{mid}(X_0)) \approx (*, 0, 0, \ldots, 0)$$

does not add relevant error to the result $Q(P(X) - x_0)$.

Flow is locally almost constant $\Longrightarrow$ $QDf(X_0)$ thin interval matrix.

Conclusion: size of Y is quadratic in set diameter

$$QDf(X_0)(X_0 - \mathrm{mid}(X_0))\Delta t$$

Locally orthogonal sections

Minimize "sliding" part:

$$\begin{aligned} Y &= \texttt{eval}(X_0, Q \circ f) \cdot \Delta t \\ &\subset (Qf(\mathrm{mid}(X_0)) + QDf(X_0)(X_0 - \mathrm{mid}(X_0)))\Delta t \end{aligned}$$

The term

$$Qf(\mathrm{mid}(X_0)) \approx (*, 0, 0, \ldots, 0)$$

does not add relevant error to the result $Q(P(X) - x_0)$.

Flow is locally almost constant $\Longrightarrow QDf(X_0)$ thin interval matrix.

Conclusion: size of Y is quadratic in set diameter

$$QDf(X_0)(X_0 - \mathrm{mid}(X_0))\Delta t$$

Locally orthogonal sections

Minimize “sliding” part:

$$\begin{aligned} Y &= \mathtt{eval}(X_0, Q \circ f) \cdot \Delta t \\ &\subset (Qf(\mathrm{mid}(X_0)) + QDf(X_0)(X_0 - \mathrm{mid}(X_0)))\Delta t \end{aligned}$$

The term

$$Qf(\mathrm{mid}(X_0)) \approx (*, 0, 0, \ldots, 0)$$

does not add relevant error to the result $Q(P(X) - x_0)$.

Flow is locally almost constant $\Longrightarrow$ $QDf(X_0)$ thin interval matrix.

Conclusion: size of Y is quadratic in set diameter

$$QDf(X_0)(X_0 - \mathrm{mid}(X_0))\Delta t$$

Locally orthogonal sections

Minimize "sliding" part:

$$\begin{aligned} Y &= \mathtt{eval}(X_0, Q \circ f) \cdot \Delta t \\ &\subset (Qf(\mathrm{mid}(X_0)) + QDf(X_0)(X_0 - \mathrm{mid}(X_0)))\Delta t \end{aligned}$$

The term

$$Qf(\mathrm{mid}(X_0)) \approx (*, 0, 0, \ldots, 0)$$

does not add relevant error to the result $Q(P(X) - x_0)$.

Flow is locally almost constant $\Longrightarrow$ $QDf(X_0)$ thin interval matrix.

Conclusion: size of Y is quadratic in set diameter

$$QDf(X_0)(X_0 - \mathrm{mid}(X_0))\Delta t$$

Data from some 8 DIM ODE

$Y0$	Y	ΔY
maxDiam=4.88e-05	maxDiam=2.62e-09	maxDiam=5.02e-9
[-2.44e-05, 2.44e-05] [-5.31e-06, 5.31e-06] [-9.46e-07, 9.46e-07] [-8.41e-10, 8.41e-10] [-2.92e-10, 2.92e-10] [-8.91e-11, 8.91e-11] [-1.82e-10, 1.82e-10] [-3.63e-11, 3.63e-11]	[-5.08e-10, 5.08e-10] [-1.15e-09, 1.15e-09] [-1.31e-09, 1.31e-09] [-9.22e-11, 9.22e-11] [-9.62e-11, 9.62e-11] [-1.49e-11, 1.49e-11] [-1.19e-11, 1.19e-11] [-5.22e-12, 5.22e-12]	[-6.58e-10, 9.60e-10] [-2.69e-11, 7.13e-10] [-3.65e-10, 1.05e-09] [-8.26e-10, 7.58e-10] [-2.11e-09, 2.17e-09] [-1.10e-09, 1.10e-09] [-2.50e-09, 2.52e-09] [-8.48e-10, 8.61e-10]

Data from some 8 DIM ODE

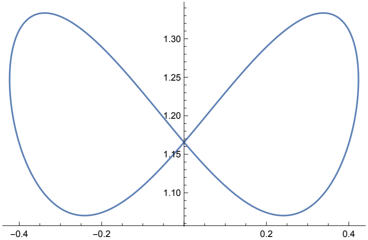
$Y0$	Y	ΔY
maxDiam=0.0005	maxDiam=3.01e-07	maxDiam=5.95e-7
[-0.00025, 0.00025]	[-5.91e-08, 5.91e-08]	[-7.80e-08, 1.10e-07]
[-5.79e-05, 5.79e-05]	[-1.32e-07, 1.32e-07]	[-3.16e-09, 8.19e-08]
[-1.38e-05, 1.38e-05]	[-1.50e-07, 1.50e-07]	[-4.20e-08, 1.20e-07]
[-9.41e-08, 9.41e-08]	[-1.09e-08, 1.09e-08]	[-9.53e-08, 8.71e-08]
[-3.23e-08, 3.23e-08]	[-1.14e-08, 1.14e-08]	[-2.48e-07, 2.64e-07]
[-9.54e-09, 9.54e-09]	[-2.03e-09, 2.03e-09]	[-1.33e-07, 1.38e-07]
[-2.00e-08, 2.00e-08]	[-1.78e-09, 1.78e-09]	[-2.88e-07, 3.07e-07]
[-3.75e-09, 3.75e-09]	[-8.04e-10, 8.04e-10]	[-1.00e-07, 1.18e-07]

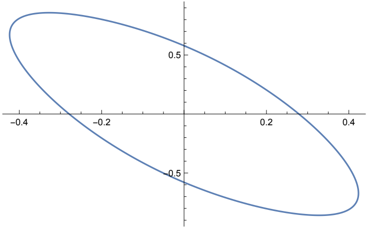
Example: periodic point for K-S PDE $u_t + \nu u_{xxxx} + u_{xx} + uu_x = 0$

Brouwer theorem:
If $P(B(u,r)) \subset B(u,r)$
then periodic point exists.

approximate periodic point:

```
long double u[] =  {
  0,
  1.16553884476,
  0.579136420845,
  -0.276891412264,
  -0.125829123416,
  0.0130156137922,
  0.0167574217536,
  0.00073178705754,
  -0.00147559942146,
  -0.000256013145596,
  9.54274711846e-05,
  3.26275575219e-05,
  -3.71643369111e-06,
  -2.98856651526e-06,
  -6.61935314369e-08,
  2.16021996994e-07};
```



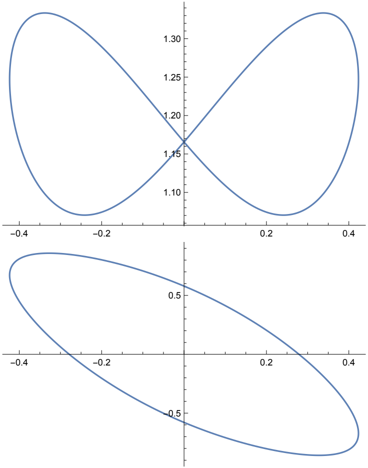


Example: periodic point for K-S PDE $u_t + \nu u_{xxxx} + u_{xx} + uu_x = 0$

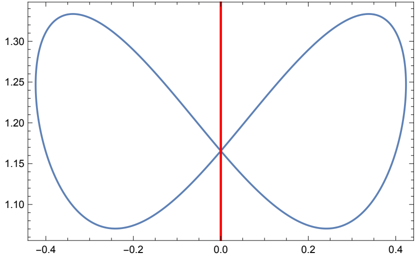
approximate periodic point:

```
long double u[] =  {
  0,
  1.16553884476,
  0.579136420845,
  -0.276891412264,
  -0.125829123416,
  0.0130156137922,
  0.0167574217536,
  0.00073178705754,
  -0.00147559942146,
  -0.000256013145596,
  9.54274711846e-05,
  3.26275575219e-05,
  -3.71643369111e-06,
  -2.98856651526e-06,
  -6.61935314369e-08,
  2.16021996994e-07};
```

Brouwer theorem:
If $P(B(u,r)) \subset B(u,r)$
then periodic point exists.



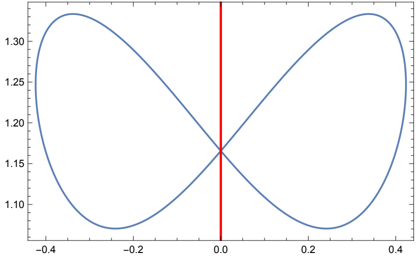
The section:
$\Pi = \{x_1 = 0\}$
Coordinates on section:
orthonormalized Jordan basis



r=1e-5, DIM=8
$P(B(u,r)) \not\subset B(u,r)$:

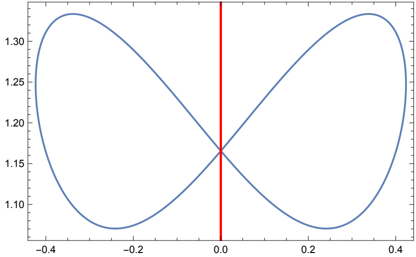
```
[-4.9409474787e-05, 4.94092032917e-05]
[-3.67513837624e-05, 3.67511583982e-05]
[-2.92501322531e-06, 2.92492222988e-06]
[-1.3185305566e-07, 1.31877347549e-07]
[-1.13187855439e-07, 1.13203957946e-07]
[-4.3150814398e-08, 4.32125456569e-08]
[-9.12539283695e-09, 9.10375795686e-09]
```

The section:
$\Pi = \{x_1 = 0\}$
Coordinates on section:
orthonormalized Jordan basis



r=1e-5, DIM=8
$P(B(u,r)) \not\subset B(u,r)$:

```
[-4.9409474787e-05, 4.94092032917e-05]
[-3.67513837624e-05, 3.67511583982e-05]
[-2.92501322531e-06, 2.92492222988e-06]
[-1.3185305566e-07, 1.31877347549e-07]
[-1.13187855439e-07, 1.13203957946e-07]
[-4.3150814398e-08, 4.32125456569e-08]
[-9.12539283695e-09, 9.10375795686e-09]
```



The section:

$\Pi = \{x_1 = 0\}$

Coordinates on section:

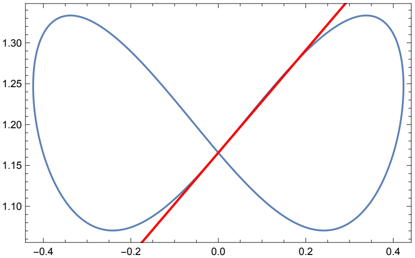
orthonormalized Jordan basis

r=1e-5, DIM=8

$P(B(u,r)) \not\subset B(u,r)$:

```
[-4.9409474787e-05, 4.94092032917e-05]
[-3.67513837624e-05, 3.67511583982e-05]
[-2.92501322531e-06, 2.92492222988e-06]
[-1.3185305566e-07, 1.31877347549e-07]
[-1.13187855439e-07, 1.13203957946e-07]
[-4.3150814398e-08, 4.32125456569e-08]
[-9.12539283695e-09, 9.10375795686e-09]
```

The section:
orthogonal at u
Coordinates on section:
orthonormalized Jordan basis

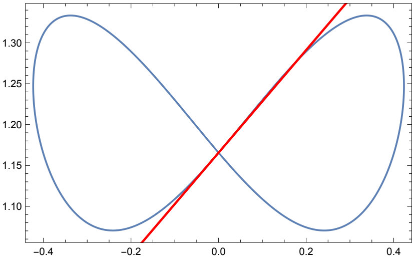


r=1e-5, DIM=8
$P(B(u,r)) \subset B(u,r)$:

```
[-5.31605396147e-06, 5.31674137118e-06]
[-9.42191230244e-07, 9.42878588724e-07]
[-1.66220842927e-09, 1.59463296395e-09]
[-2.45153462935e-09, 2.50550974418e-09]
[-1.18006303449e-09, 1.18024850156e-09]
[-2.6604271649e-09, 2.68510394746e-09]
[-8.76052652079e-10, 8.89090681198e-10]
```

Approximate leading eigenvalues of $DP(u)$:
0.526674371825, 0.0900815643755

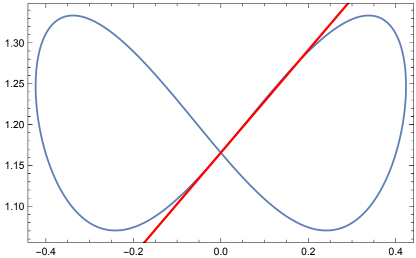
The section:
orthogonal at u
Coordinates on section:
orthonormalized Jordan basis



r=1e-5, DIM=8
$P(B(u,r)) \subset B(u,r)$:

```
[-5.31605396147e-06, 5.31674137118e-06]
[-9.42191230244e-07, 9.42878588724e-07]
[-1.66220842927e-09, 1.59463296395e-09]
[-2.45153462935e-09, 2.50550974418e-09]
[-1.18006303449e-09, 1.18024850156e-09]
[-2.6604271649e-09, 2.68510394746e-09]
[-8.76052652079e-10, 8.89090681198e-10]
```

Approximate leading eigenvalues of $DP(u)$:
0.526674371825, 0.0900815643755



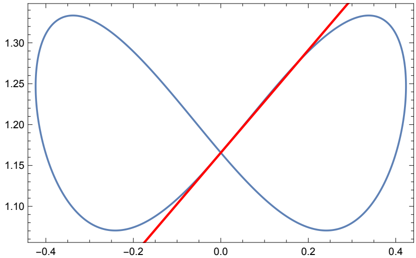
The section:
orthogonal at u
Coordinates on section:
orthonormalized Jordan basis

r=1e-5, DIM=8
$P(B(u,r)) \subset B(u,r)$:

```
[-5.31605396147e-06, 5.31674137118e-06]
[-9.42191230244e-07, 9.42878588724e-07]
[-1.66220842927e-09, 1.59463296395e-09]
[-2.45153462935e-09, 2.50550974418e-09]
[-1.18006303449e-09, 1.18024850156e-09]
[-2.6604271649e-09, 2.68510394746e-09]
[-8.76052652079e-10, 8.89090681198e-10]
```

Approximate leading eigenvalues of $DP(u)$:
0.526674371825, 0.0900815643755

The section:
orthogonal at u
Coordinates on section:
orthonormalized Jordan basis



r=1e-5, DIM=8
$P(B(u,r)) \subset B(u,r)$:

```
[-5.31605396147e-06, 5.31674137118e-06]
[-9.42191230244e-07, 9.42878588724e-07]
[-1.66220842927e-09, 1.59463296395e-09]
[-2.45153462935e-09, 2.50550974418e-09]
[-1.18006303449e-09, 1.18024850156e-09]
[-2.6604271649e-09, 2.68510394746e-09]
[-8.76052652079e-10, 8.89090681198e-10]
```

Approximate leading eigenvalues of $DP(u)$:
0.526674371825, 0.0900815643755

The section: orthogonal at u
Coordinates on section: orthonormalized Jordan basis
r=1e-5, DIM=16
$P(B(u,r)) \subset B(u,r)$:

```
[-5.30518371635e-06, 5.30452691649e-06]
[-9.42998614891e-07, 9.42340314603e-07]
[-7.1911008934e-13, 5.81538172908e-13]
[-1.56063446241e-09, 1.49626312558e-09]
[-1.82179762784e-09, 1.86019508708e-09]
[-1.80519332123e-09, 1.78782861756e-09]
[-3.34287067059e-09, 3.40090471495e-09]
[-2.24445883539e-09, 2.28533477411e-09]
[-6.83498541461e-10, 7.04685901253e-10]
[-5.56782384558e-09, 5.49302729071e-09]
[-2.83074203767e-10, 2.67469325907e-10]
[-5.91645614262e-10, 5.83804248496e-10]
[-1.46949233434e-09, 1.44558774576e-09]
[-5.93548847204e-10, 5.87910838106e-10]
[-4.26049462538e-12, 5.01778131963e-12]
```

The section: orthogonal at u

Coordinates on section: orthonormalized Jordan basis

r=1e-5, DIM=30

$P(B(u,r)) \subset B(u,r)$:

```
[-5.31139731645e-06, 5.3106576831e-06]
[-9.47614873739e-07, 9.46873610315e-07]
[-1.67870994542e-09, 1.75124066428e-09]
[-2.07442986912e-09, 2.11599482968e-09]
[-1.22123294184e-18, 1.22561422379e-18]
[-2.03677108416e-09, 2.05840349398e-09]
[-2.72922718755e-09, 2.70869294073e-09]
[-1.01440732795e-09, 1.02041496357e-09]
[-4.41659318325e-10, 4.35320365755e-10]
[-3.26251189369e-10, 3.21160585319e-10]
[-3.33208841207e-10, 3.38225115674e-10]
[-6.87207811613e-12, 6.82381698891e-12]
[-3.5839343019e-10, 3.65076264891e-10]
[-1.62372782708e-10, 1.59043253978e-10]
[-1.11677430415e-10, 1.08298764788e-10]
[-2.89625916401e-13, 2.81947740697e-13]
[-6.01913084364e-11, 6.51676540602e-11]
[-2.40173695335e-12, 2.69236343649e-12]
[-3.03383473885e-11, 2.74356986701e-11]
...
```

Part 3c

Varying section: reduce crossing time

Minimize crossing time:

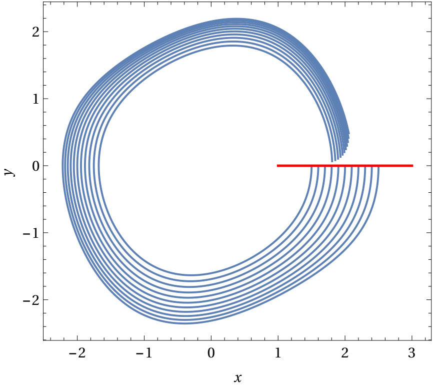
$$Y := \texttt{eval}(X_0, Q \circ f) \cdot \color{red}{\Delta t}$$
$$\Delta Y := \frac{1}{2} Q \cdot Df(E) \cdot f(E) \cdot \color{red}{\Delta t^2}$$

$t_\Pi : \Pi_1 \to \mathbb{R}$ - return time function

Observation: If

$$t_\Pi \approx \text{constant for } x \in U \subset \Pi$$

then the crossing time and estimations on P should be tighter.



Minimize crossing time:

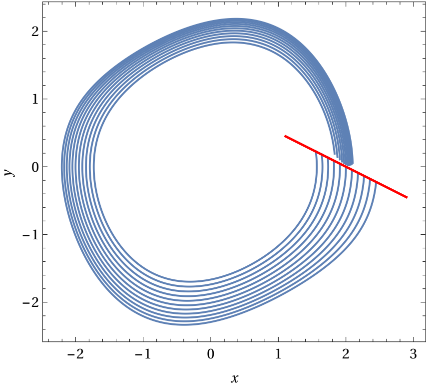
$$Y := \mathtt{eval}(X_0, Q \circ f) \cdot {\color{red}\Delta t}$$
$$\Delta Y := \frac{1}{2} Q \cdot Df(E) \cdot f(E) \cdot {\color{red}\Delta t^2}$$

$t_\Pi : \Pi_1 \to \mathbb{R}$ - return time function

Observation: If

$$t_\Pi \approx \text{constant for } x \in U \subset \Pi$$

then the crossing time and estimations on P should be tighter.



Minimize crossing time:

Case of fixed point: assume $P(x) = x$

$\alpha(x) = 0$ - defines section

$A := \frac{\partial}{\partial x}\varphi(t = t_\Pi(x), x)$

$$\alpha(\varphi(t_\Pi(x), x)) \equiv 0$$

$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T A \equiv 0$$

If $\nabla\alpha(x)$ is left eigenvector for A for $\lambda = 1$ then

$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T \frac{\partial}{\partial x} A =$$

$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x) + \nabla\alpha(x) \equiv 0$$

$$\Downarrow$$

$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear

$$\Downarrow$$

$\frac{\partial t_\Pi}{\partial v}(x) = 0$ for $v \in T_x\Pi$

Minimize crossing time:

Case of fixed point: assume $P(x) = x$
$\alpha(x) = 0$ - defines section
$A := \frac{\partial}{\partial x}\varphi(t = t_\Pi(x), x)$

$$\alpha(\varphi(t_\Pi(x), x)) \equiv 0$$
$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T A \equiv 0$$

If $\nabla\alpha(x)$ is left eigenvector for A for $\lambda = 1$ then

$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T \frac{\partial}{\partial x} A =$$
$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x) + \nabla\alpha(x) \equiv 0$$
$$\Downarrow$$
$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear
$$\Downarrow$$
$$\frac{\partial t_\Pi}{\partial v}(x) = 0 \text{ for } v \in T_x\Pi$$

Minimize crossing time:

Case of fixed point: assume $P(x) = x$
$\alpha(x) = 0$ - defines section
$A := \frac{\partial}{\partial x}\varphi(t = t_\Pi(x), x)$

$$\alpha(\varphi(t_\Pi(x), x)) \equiv 0$$
$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T A \equiv 0$$

If $\nabla\alpha(x)$ is left eigenvector for A for $\lambda = 1$ then

$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T \frac{\partial}{\partial x} A =$$
$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x) + \nabla\alpha(x) \equiv 0$$
$$\Downarrow$$
$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear
$$\Downarrow$$
$$\frac{\partial t_\Pi}{\partial v}(x) = 0 \text{ for } v \in T_x\Pi$$

Minimize crossing time:

Case of fixed point: assume $P(x) = x$
$\alpha(x) = 0$ - defines section
$A := \frac{\partial}{\partial x}\varphi(t = t_\Pi(x), x)$

$$\begin{aligned} \alpha(\varphi(t_\Pi(x), x)) &\equiv 0 \\ \langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T A &\equiv 0 \end{aligned}$$

If $\nabla\alpha(x)$ is left eigenvector for A for $\lambda = 1$ then

$$\langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x)^T + \nabla\alpha(x)^T \frac{\partial}{\partial x} A = \\ \langle \nabla\alpha(x); f(x)\rangle \nabla t_\Pi(x) + \nabla\alpha(x) \equiv 0$$

$$\Downarrow$$

$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear

$$\Downarrow$$

$\frac{\partial t_\Pi}{\partial v}(x) = 0$ for $v \in T_x\Pi$

Minimize crossing time:

Case of fixed point: assume $P(x) = x$
$\alpha(x) = 0$ - defines section
$A := \frac{\partial}{\partial x}\varphi(t = t_\Pi(x), x)$

$$\alpha(\varphi(t_\Pi(x), x)) \equiv 0$$
$$\langle\nabla\alpha(x); f(x)\rangle\nabla t_\Pi(x)^T + \nabla\alpha(x)^T A \equiv 0$$

If $\nabla\alpha(x)$ is left eigenvector for A for $\lambda = 1$ then

$$\langle\nabla\alpha(x); f(x)\rangle\nabla t_\Pi(x)^T + \nabla\alpha(x)^T \frac{\partial}{\partial x} A =$$
$$\langle\nabla\alpha(x); f(x)\rangle\nabla t_\Pi(x) + \nabla\alpha(x) \equiv 0$$

$$\Downarrow$$

$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear

$$\Downarrow$$

$\frac{\partial t_\Pi}{\partial v}(x) = 0$ for $v \in T_x\Pi$

Minimize crossing time:

Case of fixed point: assume $P(x) = x$
$\alpha(x) = 0$ - defines section
$A := \frac{\partial}{\partial x}\varphi(t = t_\Pi(x), x)$

$$\begin{aligned}\alpha(\varphi(t_\Pi(x), x)) &\equiv 0\\ \langle\nabla\alpha(x); f(x)\rangle\nabla t_\Pi(x)^T + \nabla\alpha(x)^T A &\equiv 0\end{aligned}$$

If $\nabla\alpha(x)$ is left eigenvector for A for $\lambda = 1$ then

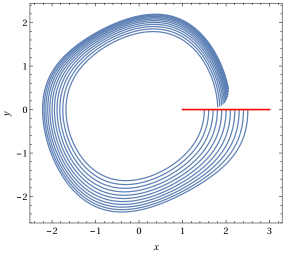
$$\langle\nabla\alpha(x); f(x)\rangle\nabla t_\Pi(x)^T + \nabla\alpha(x)^T \frac{\partial}{\partial x} A =$$
$$\langle\nabla\alpha(x); f(x)\rangle\nabla t_\Pi(x) + \nabla\alpha(x) \equiv 0$$
$$\Downarrow$$
$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear
$$\Downarrow$$
$$\frac{\partial t_\Pi}{\partial v}(x) = 0 \text{ for } v \in T_x\Pi$$

Example: van der Pol equation

Equation:

$$x'' = 0.2x'(1 - x^2) - x$$

The section: $\Pi = \{y = 0\}$ (orthogonal)



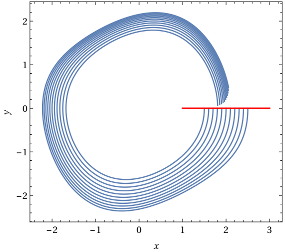
init diam	crossing time	x_0	x_1
1e-10	3.61e-11	[-3.61e-11, 3.61e-11]	[-2.83e-11, 2.83e-11]
1e-09	3.6e-10	[-3.6e-10, 3.6e-10]	[-2.83e-10, 2.83e-10]
1e-08	3.6e-09	[-3.6e-09, 3.6e-09]	[-2.83e-09, 2.83e-09]
1e-07	3.6e-08	[-3.6e-08, 3.6e-08]	[-2.83e-08, 2.83e-08]
1e-06	3.6e-07	[-3.6e-07, 3.6e-07]	[-2.83e-07, 2.83e-07]
1e-05	3.6e-06	[-3.6e-06, 3.6e-06]	[-2.83e-06, 2.83e-06]
0.0001	3.61e-05	[-3.61e-05, 3.61e-05]	[-2.83e-05, 2.83e-05]
0.001	0.000364	[-0.000364, 0.000364]	[-0.000284, 0.000284]
0.01	0.00397	[-0.00398, 0.00397]	[-0.00293, 0.00293]

Example: van der Pol equation

Equation:

$$x'' = 0.2x'(1 - x^2) - x$$

The section: $\Pi = \{y = 0\}$ (orthogonal)



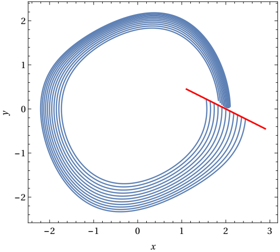
init diam	crossing time	x_0	x_1
1e-10	3.61e-11	[-3.61e-11, 3.61e-11]	[-2.83e-11, 2.83e-11]
1e-09	3.6e-10	[-3.6e-10, 3.6e-10]	[-2.83e-10, 2.83e-10]
1e-08	3.6e-09	[-3.6e-09, 3.6e-09]	[-2.83e-09, 2.83e-09]
1e-07	3.6e-08	[-3.6e-08, 3.6e-08]	[-2.83e-08, 2.83e-08]
1e-06	3.6e-07	[-3.6e-07, 3.6e-07]	[-2.83e-07, 2.83e-07]
1e-05	3.6e-06	[-3.6e-06, 3.6e-06]	[-2.83e-06, 2.83e-06]
0.0001	3.61e-05	[-3.61e-05, 3.61e-05]	[-2.83e-05, 2.83e-05]
0.001	0.000364	[-0.000364, 0.000364]	[-0.000284, 0.000284]
0.01	0.00397	[-0.00398, 0.00397]	[-0.00293, 0.00293]

Example: van der Pol equation

Equation:

$$x'' = 0.2x'(1 - x^2) - x$$

The section: minimizes crossing time



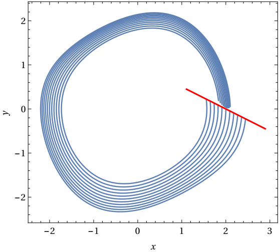
init diam	crossing time	x_0	x_1
1e-10	3.46e-14	[-2.91e-14, 2.93e-14]	[-2.83e-11, 2.83e-11]
1e-09	3.46e-14	[-2.91e-14, 2.93e-14]	[-2.83e-10, 2.83e-10]
1e-08	3.55e-14	[-2.94e-14, 2.95e-14]	[-2.83e-09, 2.83e-09]
1e-07	6.48e-14	[-5.56e-14, 5.58e-14]	[-2.83e-08, 2.83e-08]
1e-06	2.99e-12	[-2.67e-12, 2.67e-12]	[-2.83e-07, 2.83e-07]
1e-05	2.96e-10	[-2.64e-10, 2.64e-10]	[-2.83e-06, 2.83e-06]
0.0001	2.96e-08	[-2.64e-08, 2.64e-08]	[-2.83e-05, 2.83e-05]
0.001	2.97e-06	[-2.65e-06, 2.65e-06]	[-0.000284, 0.000284]
0.01	0.000311	[-0.000278, 0.000278]	[-0.003, 0.003]

Example: van der Pol equation

Equation:

$$x'' = 0.2x'(1 - x^2) - x$$

The section: minimizes crossing time



init diam	crossing time	x_0	x_1
1e-10	3.46e-14	[-2.91e-14, 2.93e-14]	[-2.83e-11, 2.83e-11]
1e-09	3.46e-14	[-2.91e-14, 2.93e-14]	[-2.83e-10, 2.83e-10]
1e-08	3.55e-14	[-2.94e-14, 2.95e-14]	[-2.83e-09, 2.83e-09]
1e-07	6.48e-14	[-5.56e-14, 5.58e-14]	[-2.83e-08, 2.83e-08]
1e-06	2.99e-12	[-2.67e-12, 2.67e-12]	[-2.83e-07, 2.83e-07]
1e-05	2.96e-10	[-2.64e-10, 2.64e-10]	[-2.83e-06, 2.83e-06]
0.0001	2.96e-08	[-2.64e-08, 2.64e-08]	[-2.83e-05, 2.83e-05]
0.001	2.97e-06	[-2.65e-06, 2.65e-06]	[-0.000284, 0.000284]
0.01	0.000311	[-0.000278, 0.000278]	[-0.003, 0.003]

Computer Assisted Proofs in Dynamics group

- Main
- Research interests
- The CAPD group
- Applications of the CAPD
- Download the library
- CAPD 4.0 Documentation
- RedHom subproject
- Related links

Contact:
Institute of Computer Science
Jagiellonian University
Lojasiewicza 6
30-348 Krakow, Poland

virtual:
wilczak@ii.uj.edu.pl

What is the CAPD library?

The CAPD library is a collection of flexible C++ modules which are mainly designed to computation of homology of sets and maps and nonrigorous and validated numerics for dynamical systems.

The list of modules is pretty long, but the most important are:

Basic modules:

- krak - a portable graphics kernel for very primitive drawing in the graphical window. Very easy to start with.
- interval - template written interval arithmetic, supports double, long double and multiprecision. It can be extended to any arithmetic type for which we can implement arithmetic operations and rounding.
- vectalg and matrixAlgorithms - a flexible template implementation of basic operations and algorithms for **dense** vectors and matrices (with integer, floating points and various interval coefficients).

Modules for dynamical systems:

- map - computation of values and derivatives of maps. It is also the core for the solvers in *dynsys* module.
- dynsys - various nonrigorous and rigorous solvers to ODEs, for computations of the solutions and partial derivatives wrt initial conditions up to arbitrary order.
- geomset, dynset - various representations of sets and Lohner-type algorithms.
- poincare - computation of Poincare maps and their derivatives; both rigorous and nonrigorous.
- diffIncl - rigorous computations of the solutions to differential inclusions.

Modules for computation of homology:

- Currently developed and recommended homological software is based on various reduction algorithms. The RedHom homology project is the official **subproject** of the CAPD library.

http://capd.ii.uj.edu.pl

Computer Assisted Proofs in Dynamics

The capdDynSys 4.0 in 2015:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ ODE solvers
- Poincaré maps and their $r-$th order derivatives
- Differential inclusions
- supports: double, long double, multiprecision, interval, mpfr-intervals

Some applications:

- $\mathcal{C}^0$-computations;

  chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$-computations;

  periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, homoclinic and heteroclinic solutions for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2$-computations;

  cocoon bifurcations, homoclinic tangencies
- $\mathcal{C}^3 - \mathcal{C}^5$ computations;

  bifurcations of periodic orbits for ODEs, KAM stability of periodic solutions, invariant tori around periodic orbits

The capdDynSys 4.0 in 2015:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ ODE solvers
- Poincaré maps and their $r-$th order derivatives
- Differential inclusions
- supports: double, long double, multiprecision, interval, mpfr-intervals

Some applications:

- $\mathcal{C}^0$-computations;

  chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$-computations;

  periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, homoclinic and heteroclinic solutions for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2$-computations;

  cocoon bifurcations, homoclinic tangencies
- $\mathcal{C}^3 - \mathcal{C}^5$ computations;

  bifurcations of periodic orbits for ODEs, KAM stability of periodic solutions, invariant tori around periodic orbits

The capdDynSys 4.0 in 2015:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ ODE solvers
- Poincaré maps and their $r-$th order derivatives
- Differential inclusions
- supports: double, long double, multiprecision, interval, mpfr-intervals

Some applications:

- $\mathcal{C}^0$-computations;

  chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$-computations;

  periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, homoclinic and heteroclinic solutions for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2$-computations;

  cocoon bifurcations, homoclinic tangencies
- $\mathcal{C}^3 - \mathcal{C}^5$ computations;

  bifurcations of periodic orbits for ODEs, KAM stability of periodic solutions, invariant tori around periodic orbits

The capdDynSys 4.0 in 2015:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ ODE solvers
- Poincaré maps and their $r-$th order derivatives
- Differential inclusions
- supports: double, long double, multiprecision, interval, mpfr-intervals

Some applications:

- $\mathcal{C}^0$-computations;

  chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$-computations;

  periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, homoclinic and heteroclinic solutions for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2$-computations;

  cocoon bifurcations, homoclinic tangencies
- $\mathcal{C}^3 - \mathcal{C}^5$ computations;

  bifurcations of periodic orbits for ODEs, KAM stability of periodic solutions, invariant tori around periodic orbits

The capdDynSys 4.0 in 2015:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ ODE solvers
- Poincaré maps and their $r-$th order derivatives
- Differential inclusions
- supports: double, long double, multiprecision, interval, mpfr-intervals

Some applications:

- $\mathcal{C}^0$-computations;

  chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$-computations;

  periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, homoclinic and heteroclinic solutions for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2$-computations;

  cocoon bifurcations, homoclinic tangencies
- $\mathcal{C}^3 - \mathcal{C}^5$ computations;

  bifurcations of periodic orbits for ODEs, KAM stability of periodic solutions, invariant tori around periodic orbits

The capdDynSys 4.0 in 2015:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ ODE solvers
- Poincaré maps and their $r-$th order derivatives
- Differential inclusions
- supports: double, long double, multiprecision, interval, mpfr-intervals

Some applications:

- $\mathcal{C}^0$-computations;

  chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$-computations;

  periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, homoclinic and heteroclinic solutions for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2$-computations;

  cocoon bifurcations, homoclinic tangencies
- $\mathcal{C}^3 - \mathcal{C}^5$ computations;

  bifurcations of periodic orbits for ODEs, KAM stability of periodic solutions, invariant tori around periodic orbits

Thank you for your attention

Děkuji vám za pozornost