Computer-assisted proofs in dynamics
Part I: topological methods

Daniel Wilczak

Institute of Computer Science and Computational Mathematics
Jagiellonian University, Kraków, Poland

June 19, 2016

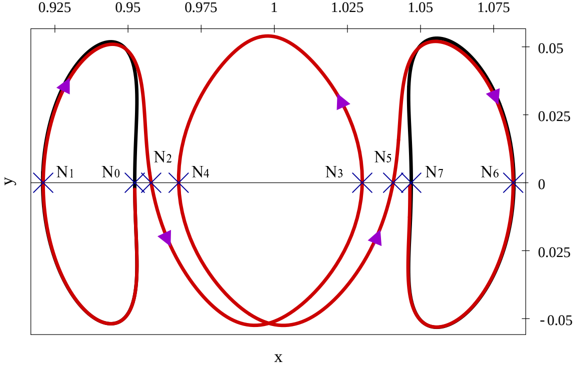
Some computer assisted proofs in dynamics

- **Langford, 1982**
  proof of Feigenbaum universality conjectures
- **Eckmann, Koch, Wittwer, 1984**
  universality for area-preserving maps
- **Grebogi, Hammel, Yorke, 1987**
  rigorous numerical shadowing of trajectories
- **Neumaier, Rage, Schlier, 1994**
  chaos in the molecular Thiele-Wilson model
- **Mischaikow and Mrozek, 1995**
  chaos in Lorenz equations
- **Mischaikow and Mrozek, 1995**
  chaos in Lorenz equations
- **Zgliczyński, 1997**
  chaos in the Hénon map and in the Rössler system
- **Palmer, Coomes, Kocak, Stoffer, Kirchgraber, 1996-2003**
  chaos via shadowing for Henon map, PCR3BP
- **W. Tucker, 2001**
  geometric model for Lorenz attractor

PROBLEM TO SOLVE:

prove that a system has solutions satisfying certain properties

- periodic solutions
- connecting orbits
- two-point boundary value problem
- invariant manifold of equilibrium, periodic orbit,
- chaos
- attractors, ...



Lyapunov orbits and heteroclinic connection in the PCR3BP

Computer-assisted proofs - methodologies

APPROACH 1: transform the problem to

$$\mathcal{F}(\mathbf{u}) = 0$$

and solve it by (interval) Newton-like scheme

Pros:

- powerful in low-dimensional discrete dynamical systems
- sometimes the only (known to me) approach that works
  (parametrization method for invariant tori)
- high accuracy

Difficulties:

- **often problem dependent**
  (especially in continuous-time systems)
- infinite dimension: the unknown $\mathbf{u}$ is often a function
  (even if the phase-space of the system is low-dimensional)
- rather difficult to handle global dynamics

Computer-assisted proofs - methodologies

APPROACH 1: transform the problem to

$$\mathcal{F}(\mathbf{u}) = 0$$

and solve it by (interval) Newton-like scheme

Pros:

- powerful in low-dimensional discrete dynamical systems
- sometimes the only (known to me) approach that works

  (parametrization method for invariant tori)
- high accuracy

Difficulties:

- **often problem dependent**

  (especially in continuous-time systems)
- infinite dimension: the unknown $\mathbf{u}$ is often a function

  (even if the phase-space of the system is low-dimensional)
- rather difficult to handle global dynamics

APPROACH 2: use isolation – analyse the vector field

Pros:

- no need to integrate ODE or PDE
- works directly in the phase space
- focuses on bounded solutions
  (may work even if most of the trajectories escape to infinity)

Difficulties:

- constructing isolating blocks (segments) may be difficult
  (possible memory and CPU issues, parallelization possible)
- requires hyperbolic-like dynamics
  (rather no chance to handle elliptic solutions without extra tools)
- low accuracy - up to the isolating set

Computer-assisted proofs - methodologies

APPROACH 2: use isolation – analyse the vector field

Pros:

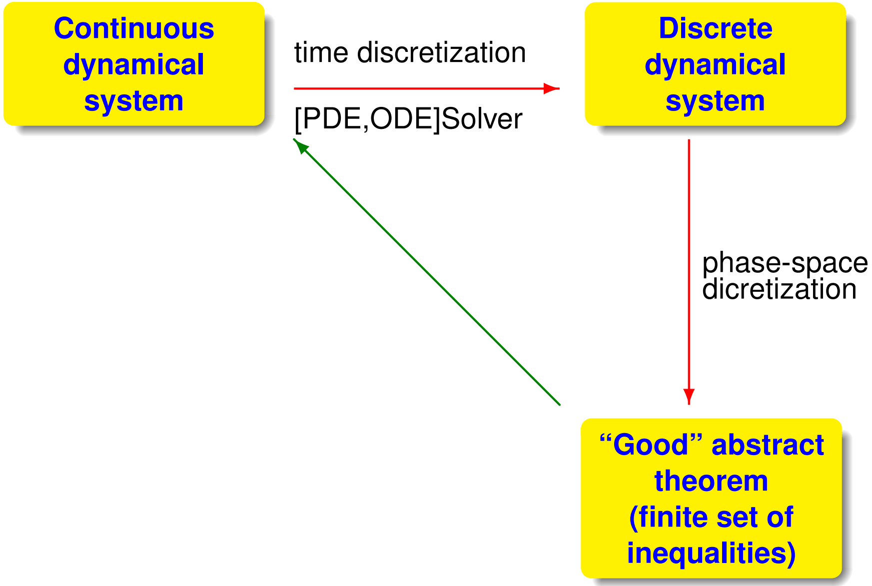
- no need to integrate ODE or PDE
- works directly in the phase space
- focuses on bounded solutions
  (may work even if most of the trajectories escape to infinity)

Difficulties:

- constructing isolating blocks (segments) may be difficult
  (possible memory and CPU issues, parallelization possible)
- requires hyperbolic-like dynamics
  (rather no chance to handle elliptic solutions without extra tools)
- low accuracy - up to the isolating set

Computer-assisted proofs - methodologies

APPROACH 3: discretize



APPROACH 3: discretize

Pros:

- rather general (problem independent)
- low-dimensional (works in the phase space of the system)
- ready to handle global dynamics
- well suited for smooth methods

Difficulties:

- integration of ODEs and PDEs is not easy
  (usually CPU issues)
- stiffness: does not work if most of solutions escape to infinity in short time
- solutions of the system may not exist (PDEs) even if the solutions we are interested in do exist
- unable to handle some types of dynamical objects
  (like invariant tori)

Computer-assisted proofs - methodologies

APPROACH 3: discretize

Pros:

- rather general (problem independent)
- low-dimensional (works in the phase space of the system)
- ready to handle global dynamics
- well suited for smooth methods

Difficulties:

- integration of ODEs and PDEs is not easy
  (usually CPU issues)
- stiffness: does not work if most of solutions escape to infinity in short time
- solutions of the system may not exist (PDEs) even if the solutions we are interested in do exist
- unable to handle some types of dynamical objects
  (like invariant tori)

Outline of Part I:

1. ODE solvers (short overview of selected methods)
2. Algorithm for computation of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study: “easy” computer-assisted proofs of
   - apparently attracting periodic orbit
   - symmetric non-attracting periodic orbit
   - existence of an attractor in the Rössler system
5. Some “good” abstract theorems and their applications:
   - chaos in the Rössler system (case study)
   - Σ_4 chaos in the Michelson system
   - Shilnikov homoclinic orbits
   - Bykov heteroclinic cycles

Outline of Part I:

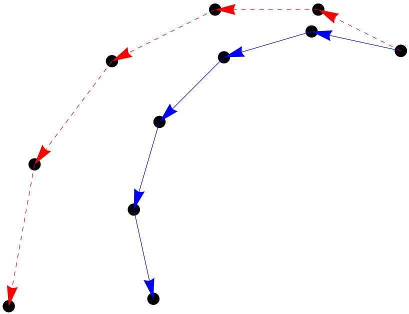
1. ODE solvers (short overview of selected methods)
2. Algorithm for computation of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study: “easy” computer-assisted proofs of
   - apparently attracting periodic orbit
   - symmetric non-attracting periodic orbit
   - existence of an attractor in the Rössler system
5. Some “good” abstract theorems and their applications:
   - chaos in the Rössler system (case study)
   - Σ_4 chaos in the Michelson system
   - Shilnikov homoclinic orbits
   - Bykov heteroclinic cycles

Outline of Part I:

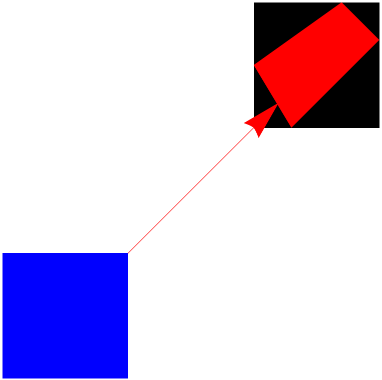
1. ODE solvers (short overview of selected methods)
2. Algorithm for computation of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study: "easy" computer-assisted proofs of
   - apparently attracting periodic orbit
   - symmetric non-attracting periodic orbit
   - existence of an attractor in the Rössler system
5. Some "good" abstract theorems and their applications:
   - chaos in the Rössler system (case study)
   - Σ_4 chaos in the Michelson system
   - Shilnikov homoclinic orbits
   - Bykov heteroclinic cycles

ODE Solvers

ODE Solvers



exact trajectory
approximate trajectory



set of initial conditions
exact image of the blue set after some time T
enclosure of the image returned by a rigorous ODE solver

ODE Solvers

Notation:

$$\begin{aligned} x' &= f(x) && \text{vector field} \\ (t,x) &\mapsto \varphi(t,x) && \text{local flow} \end{aligned}$$

INPUT:

- $[X]$ – set of initial conditions
  (interval vector, zonotope, Taylor model)
- $\widehat{h} > 0$ – a time step we want to make
  (from prediction)

OUTPUT:

- $0 < h \leq \widehat{h}$ - accepted time step
- $[Y]$ – a set
- $[X_1]$ – a set such that

$$\begin{aligned} \varphi([0,h],[X]) &\subset [Y], \\ \varphi(h,[X]) &\subset [X_1] \end{aligned}$$

ODE Solvers

Notation:

$$
\begin{aligned}
x' &= f(x) && \text{vector field}\\
(t,x) &\mapsto \varphi(t,x) && \text{local flow}
\end{aligned}
$$

INPUT:

- $[X]$ – set of initial conditions
  (interval vector, zonotope, Taylor model)
- $\widehat{h} > 0$ – a time step we want to make
  (from prediction)

OUTPUT:

- $0 < h \leq \widehat{h}$ - accepted time step
- $[Y]$ – a set
- $[X_1]$ – a set such that

$$
\begin{aligned}
\varphi([0,h],[X]) &\subset [Y],\\
\varphi(h,[X]) &\subset [X_1]
\end{aligned}
$$

ODE Solvers

Notation:

$$\begin{aligned} x' &= f(x) && \text{vector field} \\ (t,x) &\mapsto \varphi(t,x) && \text{local flow} \end{aligned}$$

INPUT:

- $[X]$ – set of initial conditions
  (interval vector, zonotope, Taylor model)
- $\widehat{h} > 0$ – a time step we want to make
  (from prediction)

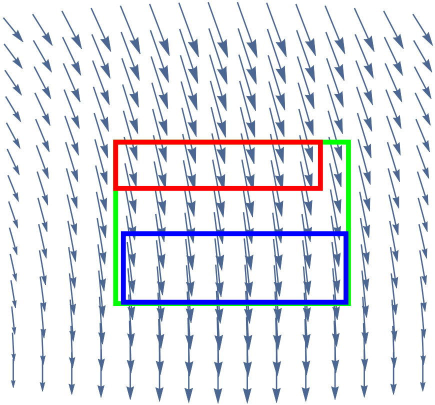
OUTPUT:

- $0 < h \leq \widehat{h}$ - accepted time step
- $[Y]$ – a set
- $[X_1]$ – a set such that

$$\begin{aligned} \varphi([0,h],[X]) &\subset [Y], \\ \varphi(h,[X]) &\subset [X_1] \end{aligned}$$

ODE Solvers

[**X**] – set of initial conditions
[$\mathbf{X_1}$] – bound for $\varphi(h, [\mathbf{X}])$
[**Y**] – bound for $\varphi([0, h], [\mathbf{X}])$



ODE Solvers - First Order Enclosure (FOE)

Theorem

$x' = f(x)$ - *ODE*, $f : \mathbb{R}^n \to \mathbb{R}^n$ – *smooth*
$[X], [Y]$ - *convex, compact sets*
$h > 0$

If

$$[Z] := [X] + [0, h]f([Y]) \subset \mathrm{int}([Y])$$

then

- $\varphi(t, x)$ *is well defined for* $(t, x) \in [0, h] \times [X]$
- *for* $t \in [0, h]$, $x \in [X]$ *there holds* $\varphi(t, x) \in [Z]$

Important prediction of h and $[Y]$

ODE Solvers - First Order Enclosure (FOE)

Theorem

$x' = f(x)$ - ODE, $f: \mathbb{R}^n \to \mathbb{R}^n$ – smooth
$[X]$, $[Y]$ - convex, compact sets
$h > 0$

If

$$[Z] := [X] + [0,h]f([Y]) \subset \mathrm{int}([Y])$$

then

- *$\varphi(t,x)$ is well defined for $(t,x) \in [0,h] \times [X]$*
- *for $t \in [0,h]$, $x \in [X]$ there holds $\varphi(t,x) \in [Z]$*

Important prediction of h and $[Y]$

Theorem

$x' = f(x)$ - ODE, $f : \mathbb{R}^n \to \mathbb{R}^n$ – smooth
$[X]$, $[Y]$ - convex, compact sets
$h > 0$

If

$$[Z] := [X] + [0, h]f([Y]) \subset \mathrm{int}([Y])$$

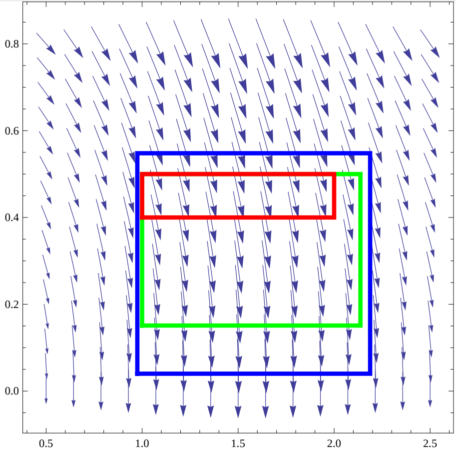
then

- *$\varphi(t, x)$ is well defined for $(t, x) \in [0, h] \times [X]$*
- *for $t \in [0, h]$, $x \in [X]$ there holds $\varphi(t, x) \in [Z]$*

Important prediction of h and $[Y]$

Example

$$x'' = -\sin(x) + 0.1x', \qquad h = 0.25$$



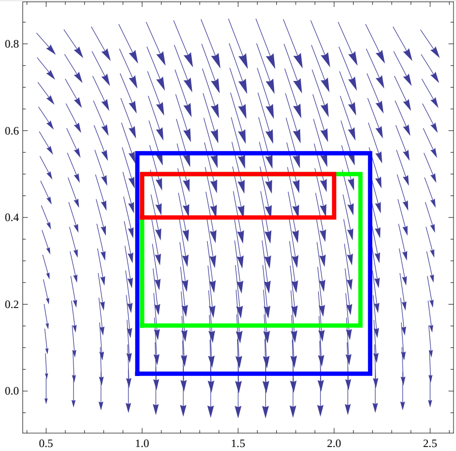
$$[\mathbf{X}] = [1,2] \times [0.4, 0.5]$$

$$[\mathbf{Y}] = [\mathbf{X}] + h[-.2, 1.5] * f([\mathbf{X}]) \subset [0.9749, 2.1875] \times [0.04, 0.548]$$

$$[\mathbf{Z}] = [\mathbf{X}] + [0, h] * f([\mathbf{Y}]) \subset [1.0, 2.137] \times [0.1502, 0.5] \subset \mathrm{int}([\mathbf{Y}])$$

Example

$$x'' = -\sin(x) + 0.1x', \qquad h = 0.25$$



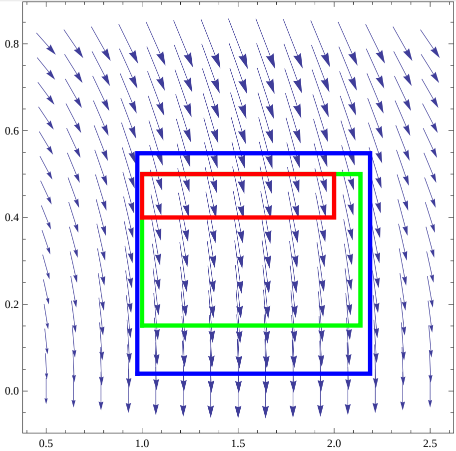
$$[\mathbf{X}] = [1, 2] \times [0.4, 0.5]$$

$$[\mathbf{Y}] = [\mathbf{X}] + h[-.2, 1.5] * f([\mathbf{X}]) \subset [0.9749, 2.1875] \times [0.04, 0.548]$$

$$[\mathbf{Z}] = [\mathbf{X}] + [0, h] * f([\mathbf{Y}]) \subset [1.0, 2.137] \times [0.1502, 0.5] \subset \operatorname{int}([\mathbf{Y}])$$

Example

$$x'' = -\sin(x) + 0.1x', \qquad h = 0.25$$



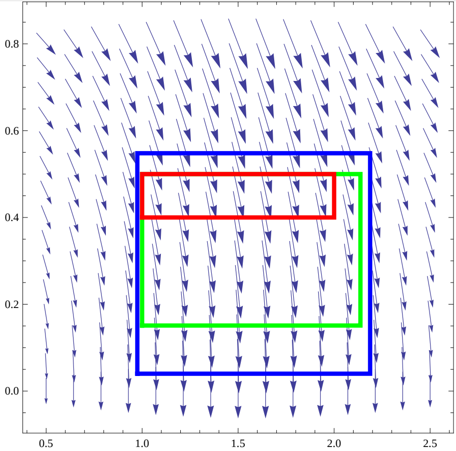
$$[\mathbf{X}] = [1, 2] \times [0.4, 0.5]$$

$$[\mathbf{Y}] = [\mathbf{X}] + h[-.2, 1.5] * f([\mathbf{X}]) \subset [0.9749, 2.1875] \times [0.04, 0.548]$$

$$[\mathbf{Z}] = [\mathbf{X}] + [0, h] * f([\mathbf{Y}]) \subset [1.0, 2.137] \times [0.1502, 0.5] \subset \mathrm{int}([\mathbf{Y}])$$

Example

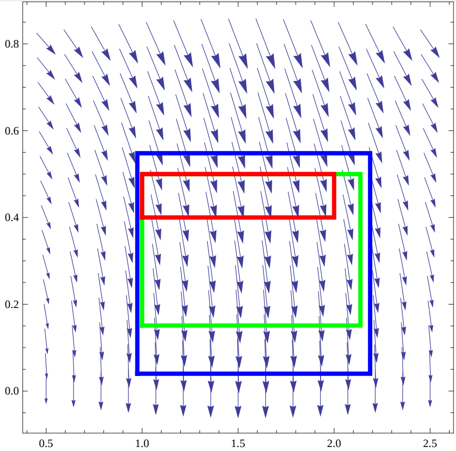
$$x'' = -\sin(x) + 0.1x', \qquad h = 0.25$$



$$
\begin{aligned}
[\mathbf{X}] &= [1,2] \times [0.4, 0.5] \\
[\mathbf{Y}] &= [\mathbf{X}] + h[-.2, 1.5] * f([\mathbf{X}]) \subset [0.9749, 2.1875] \times [0.04, 0.548] \\
[\mathbf{Z}] &= [\mathbf{X}] + [0, h] * f([\mathbf{Y}]) \subset [1.0, 2.137] \times [0.1502, 0.5] \subset \mathrm{int}([\mathbf{Y}])
\end{aligned}
$$

Example

$$x'' = -\sin(x) + 0.1x', \qquad h = 0.25$$



$$\begin{aligned}
[\mathbf{X}] &= [1,2] \times [0.4, 0.5] \\
[\mathbf{Y}] &= [\mathbf{X}] + h[-.2, 1.5] * f([\mathbf{X}]) \subset [0.9749, 2.1875] \times [0.04, 0.548] \\
[\mathbf{Z}] &= [\mathbf{X}] + [0, h] * f([\mathbf{Y}]) \subset [1.0, 2.137] \times [0.1502, 0.5] \subset \mathrm{int}([\mathbf{Y}])
\end{aligned}$$

ODE Solvers - High Order Enclosure (HOE)

Notation:
$x^{[k]}(t_0)$ – vector of k^{th} Taylor coefficients of $t \to x(t)$ at $t = t_0$

$[X]^{[k]}$ – bound for $x^{[k]}(0)$ for $x \in [X]$.

Theorem (Nedialkov, Jackson, Pryce)

$[X]$, $[R]$ - convex, compact sets
$h > 0$

$$\begin{aligned} [P] &:= [X] + [0,h][X]^{[1]} + \cdots + [0,h]^r [X]^{[r]} \\ [Y] &:= [P] + [R] \end{aligned}$$

If

$$[Z] := [0,h]^{r+1}[Y]^{[r+1]} \subset [R]$$

then

- *$\varphi(t,x)$ is well defined for $(t,x) \in [0,h] \times [X]$*
- *for $t \in [0,h]$, $x \in [X]$ there holds $\varphi(t,x) \in \sum_{i=0}^{r} t^i [X]^{[i]} + [Z]$*

ODE Solvers - High Order Enclosure (HOE)

Notation:

$x^{[k]}(t_0)$ – vector of k^{th} Taylor coefficients of $t \to x(t)$ at $t = t_0$

$[X]^{[k]}$ – bound for $x^{[k]}(0)$ for $x \in [X]$.

Theorem (Nedialkov, Jackson, Pryce)

$[X]$, $[R]$ - *convex, compact sets*
$h > 0$

$$
\begin{aligned}
[P] &:= [X] + [0,h][X]^{[1]} + \cdots + [0,h]^r[X]^{[r]} \\
[Y] &:= [P] + [R]
\end{aligned}
$$

If

$$[Z] := [0,h]^{r+1}[Y]^{[r+1]} \subset [R]$$

then

- $\varphi(t,x)$ *is well defined for* $(t,x) \in [0,h] \times [X]$
- *for* $t \in [0,h]$, $x \in [X]$ *there holds* $\varphi(t,x) \in \sum_{i=0}^{r} t^i [X]^{[i]} + [Z]$

ODE Solvers - High Order Enclosure (HOE)

Notation:

$x^{[k]}(t_0)$ – vector of k^{th} Taylor coefficients of $t \to x(t)$ at $t = t_0$

$[X]^{[k]}$ – bound for $x^{[k]}(0)$ for $x \in [X]$.

Theorem (Nedialkov, Jackson, Pryce)

$[X]$, $[R]$ - convex, compact sets

$h > 0$

$$\begin{aligned} [P] &:= [X] + [0,h][X]^{[1]} + \cdots + [0,h]^r [X]^{[r]} \\ [Y] &:= [P] + [R] \end{aligned}$$

If

$$[Z] := [0,h]^{r+1}[Y]^{[r+1]} \subset [R]$$

then

- *$\varphi(t,x)$ is well defined for $(t,x) \in [0,h] \times [X]$*
- *for $t \in [0,h]$, $x \in [X]$ there holds $\varphi(t,x) \in \sum_{i=0}^{r} t^i [X]^{[i]} + [Z]$*

Notation:

$x^{[k]}(t_0)$ – vector of k^{th} Taylor coefficients of $t \to x(t)$ at $t = t_0$

$[X]^{[k]}$ – bound for $x^{[k]}(0)$ for $x \in [X]$.

Theorem (Nedialkov, Jackson, Pryce)

$[X]$, $[R]$ - convex, compact sets

$h > 0$

$$\begin{aligned} [P] &:= [X] + [0,h][X]^{[1]} + \cdots + [0,h]^r [X]^{[r]} \\ [Y] &:= [P] + [R] \end{aligned}$$

If

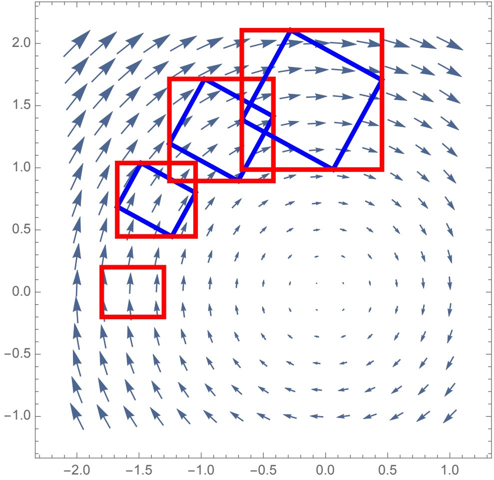
$$[Z] := [0,h]^{r+1}[Y]^{[r+1]} \subset [R]$$

then

- $\varphi(t,x)$ *is well defined for* $(t,x) \in [0,h] \times [X]$
- *for* $t \in [0,h]$, $x \in [X]$ *there holds* $\varphi(t,x) \in \sum_{i=0}^{r} t^i [X]^{[i]} + [Z]$

Example (Oscillator)

$$x'' = -x$$



ODE Solvers - wrapping effect

General idea

Propagate “good” coordinate system along with the set

- Affine representation

$$[X] = x_0 + A[r_0]$$

- Doubleton representation

$$[X] = x_0 + C[r_0] + B[r]$$

- Tripleton representation

$$[X] = x_0 + C[r_0] + Q[q] \cap B[r]$$

- Taylor Model representation (P- double coefficient sparse polynomial)

$$[X] = P(B_r) + [R]$$

ODE Solvers - wrapping effect

Φ - numerical method

$\varphi(h, x) \in \Phi(h, x) + [R]$

Reduce wrapping effect

Use mean value form: $x_0 \in [X]$

$$[X](h) \subset \Phi(h, x_0) + D_x\Phi(h, [X])([X] - x_0) + [R]$$

ODE Solvers - wrapping effect

Φ - numerical method

$\varphi(h, x) \in \Phi(h, x) + [R]$

Reduce wrapping effect

Use mean value form: $x_0 \in [X]$

$$[X](h) \subset \Phi(h, x_0) + D_x\Phi(h, [X])([X] - x_0) + [R]$$

ODE Solvers - wrapping effect

Example (Doubleton representation $[X] = x_0 + C[r_0] + B[r]$)

$$\varphi(h,[X]) \quad \subset \quad x_1 + C_1[r_0] + B_1[r_1]$$

where

$$\begin{aligned}
x_1 &= \operatorname{mid}\left(\Phi(h,x_0) + [R]\right) \\
C_1 &= \operatorname{mid}\left(D_x\Phi(h,[X]) \cdot C\right) \\
B_1 &= \text{some invertible point matrix} \\
r_1 &= \left(B_1^{-1} D_x\Phi(h,[X]) \cdot B\right)[r] + B_1^{-1}\left(\Phi(h,x_0) + [R] - x_1\right) \\
&+ B_1^{-1}\left(D_x\Phi(h,[X]) \cdot C - C_1\right)[r]
\end{aligned}$$

Choice of B_1

There are various strategies for choosing B_1

ODE Solvers - wrapping effect

Example (Doubleton representation $[X] = x_0 + C[r_0] + B[r]$)

$$\varphi(h,[X]) \subset x_1 + C_1[r_0] + B_1[r_1]$$

where

$$\begin{aligned}
x_1 &= \operatorname{mid}\left(\Phi(h,x_0) + [R]\right) \\
C_1 &= \operatorname{mid}\left(D_x\Phi(h,[X]) \cdot C\right) \\
B_1 &= \text{some invertible point matrix} \\
r_1 &= \left(B_1^{-1} D_x\Phi(h,[X]) \cdot B\right)[r] + B_1^{-1}\left(\Phi(h,x_0) + [R] - x_1\right) \\
&+ B_1^{-1}\left(D_x\Phi(h,[X]) \cdot C - C_1\right)[r]
\end{aligned}$$

Choice of B_1

There are various strategies for choosing B_1

Example (Doubleton representation $[X] = x_0 + C[r_0] + B[r]$)

$$\varphi(h,[X]) \quad \subset \quad x_1 + C_1[r_0] + B_1[r_1]$$

where

$$\begin{aligned} x_1 &= \operatorname{mid}\left(\Phi(h,x_0) + [R]\right) \\ C_1 &= \operatorname{mid}\left(D_x\Phi(h,[X]) \cdot C\right) \\ B_1 &= \text{some invertible point matrix} \\ r_1 &= \left(B_1^{-1} D_x\Phi(h,[X]) \cdot B\right)[r] + B_1^{-1}\left(\Phi(h,x_0) + [R] - x_1\right) \\ &+ B_1^{-1}\left(D_x\Phi(h,[X]) \cdot C - C_1\right)[r] \end{aligned}$$

Choice of B_1

There are various strategies for choosing B_1

What about Φ in

$$\varphi(h,[X]) \subset \Phi(h,[X]) + [R]$$

Taylor method

(arbitrary order, easy computation of $[R]$)

$$\Phi(h,x) = \sum_{k=0}^{r} h^k x^{[k]}$$

Hermite-Obreshkov method (Nedialkov, Jackson)

(implicit, arbitrary order, easy computation of $[R]$ - smaller than in Taylor method)

Put

$$\Psi_{q,p}(h,x) \quad := \quad \sum_{k=0}^{q} \binom{p+q-k}{p} / \binom{p+q}{p} h^k x^{[k]}$$

Then

$$\Psi_{q,p}(-h,x(h)) - \Psi_{p,q}(h,x) \in [R_{HO}]$$

What about Φ in

$$\varphi(h,[X]) \subset \Phi(h,[X]) + [R]$$

Taylor method

(arbitrary order, easy computation of $[R]$)

$$\Phi(h,x) = \sum_{k=0}^{r} h^k x^{[k]}$$

Hermite-Obreshkov method (Nedialkov, Jackson)

(implicit, arbitrary order, easy computation of $[R]$ - smaller than in Taylor method)

Put

$$\Psi_{q,p}(h,x) := \sum_{k=0}^{q} \binom{p+q-k}{p} / \binom{p+q}{p} h^k x^{[k]}$$

Then

$$\Psi_{q,p}(-h,x(h)) - \Psi_{p,q}(h,x) \in [R_{HO}]$$

What about Φ in

$$\varphi(h,[X]) \subset \Phi(h,[X]) + [R]$$

Taylor method

(arbitrary order, easy computation of $[R]$)

$$\Phi(h,x) = \sum_{k=0}^{r} h^k x^{[k]}$$

Hermite-Obreshkov method (Nedialkov, Jackson)

(implicit, arbitrary order, easy computation of $[R]$ - smaller than in Taylor method)

Put

$$\Psi_{q,p}(h,x) \quad := \quad \sum_{k=0}^{q} \binom{p+q-k}{p} / \binom{p+q}{p} h^k x^{[k]}$$

Then

$$\Psi_{q,p}(-h,x(h)) - \Psi_{p,q}(h,x) \in [R_{HO}]$$

What about Φ in

$$\varphi(h,[X]) \subset \Phi(h,[X]) + [R]$$

Taylor method

(arbitrary order, easy computation of $[R]$)

$$\Phi(h,x) = \sum_{k=0}^{r} h^k x^{[k]}$$

Hermite-Obreshkov method (Nedialkov, Jackson)

(implicit, arbitrary order, easy computation of $[R]$ - smaller than in Taylor method)

Put

$$\Psi_{q,p}(h,x) \quad := \quad \sum_{k=0}^{q} \binom{p+q-k}{p} / \binom{p+q}{p} h^k x^{[k]}$$

Then

$$\Psi_{q,p}(-h,x(h)) - \Psi_{p,q}(h,x) \in [R_{HO}]$$

ODE Solvers - numerical method

What about Φ in

$$\varphi(h,[X]) \subset \Phi(h,[X]) + [R]$$

Taylor method

(arbitrary order, easy computation of $[R]$)

$$\Phi(h,x) = \sum_{k=0}^{r} h^k x^{[k]}$$

Hermite-Obreshkov method (Nedialkov, Jackson)

(implicit, arbitrary order, easy computation of $[R]$ - smaller than in Taylor method)

Put

$$\Psi_{q,p}(h,x) \quad := \quad \sum_{k=0}^{q} \binom{p+q-k}{p} \Big/ \binom{p+q}{p} h^k x^{[k]}$$

Then

$$\Psi_{q,p}(-h,x(h)) - \Psi_{p,q}(h,x) \in [R_{HO}]$$

Poincaré maps

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

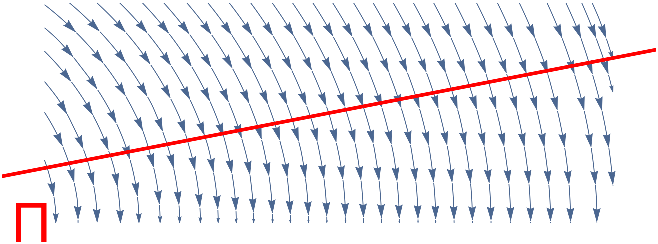
Definition

$\Pi \subset \mathbb{R}^n$ 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

$$\langle f(x); n_S(x) \rangle \neq 0$$

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*

$$\langle f(x); n_S(x) \rangle \neq 0$$

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

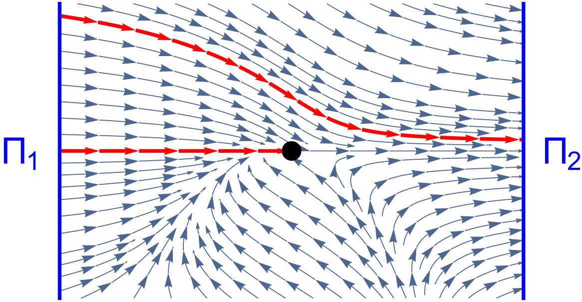
- $x \in \mathrm{dom}(P)$ iff $\varphi(t, x) \in \Pi_2$ for some $t > 0$
- $P(x)$ - first cut of $\varphi(t, x)$ with Π_2 for $t > 0$

Π_1, Π_2 - Poincaré sections for φ

Definition

$P: \Pi_1 \to \Pi_2$ - 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

- $\alpha : \mathbb{R}^n \to \mathbb{R}$ - 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

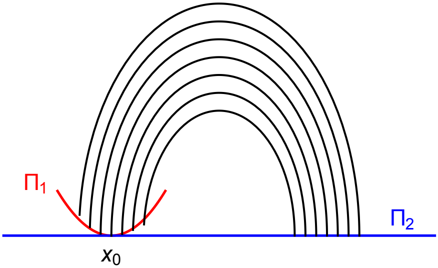
- Π_1, Π_2 - sections given by $\alpha_i : \mathbb{R}^n \to \mathbb{R}$
- $P \colon \Pi_1 \to \Pi_2$ - Poincaré map

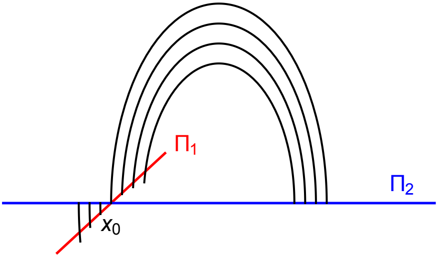
Question: is P continuous? smooth?

Settings

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

Question: is P continuous? smooth?





Theorem

Assume

- $\Pi_i = \mathrm{cl}(\mathrm{int}\Pi_i)$
- *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

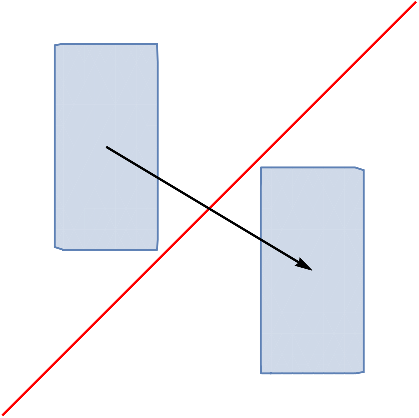
$$P(x) \in \mathrm{int}\Pi_2.$$

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

Constrains:

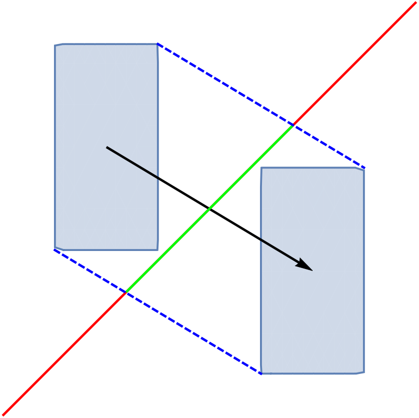
- avoid subdivisions when crossing section
- reduce wrapping effect that may occur when change representation of a set to coordinates on section



Enclosing Poincaré maps

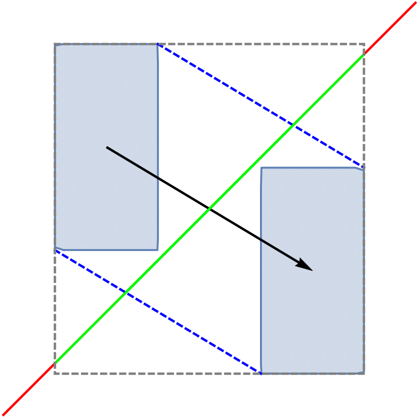
Constrains:

- avoid subdivisions when crossing section
- reduce wrapping effect that may occur when change representation of a set to coordinates on section



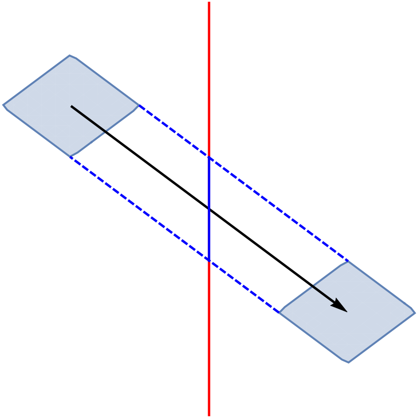
Constrains:

- avoid subdivisions when crossing section
- reduce wrapping effect that may occur when change representation of a set to coordinates on section



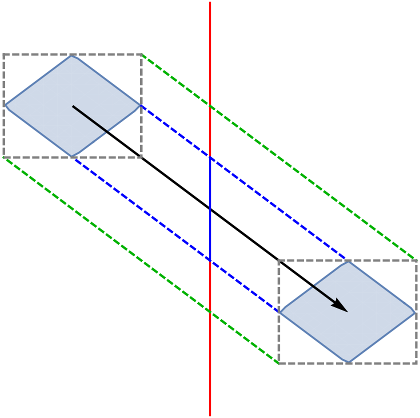
Very important:

take into account internal representation of solutions in ODE solver



Very important:

take into account internal representation of solutions in ODE solver



Abstract data structure: `RepresentableSet`

Example:

$$[X] = x + C[r_0] + B[r]$$

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:

$$Q(x - x_0 + C[r_0] + B[r]) \cap (Q(x - x_0) + (QC)[r_0] + (QB)[r])$$

Abstract data structure: `RepresentableSet`

Example:

$$[X] = x + C[r_0] + B[r]$$

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:

$$Q(x - x_0 + C[r_0] + B[r]) \cap (Q(x - x_0) + (QC)[r_0] + (QB)[r])$$

Abstract data structure: `RepresentableSet`

Example:

$$[X] = x + C[r_0] + B[r]$$

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:

$$Q(x - x_0 + C[r_0] + B[r]) \cap (Q(x - x_0) + (QC)[r_0] + (QB)[r])$$

Abstract data structure: `RepresentableSet`

Example:

$$[X] = x + C[r_0] + B[r]$$

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:

$$Q(x - x_0 + C[r_0] + B[r]) \cap (Q(x - x_0) + (QC)[r_0] + (QB)[r])$$

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: $x + C[r_0] + B[r] \subset \mathbb{R}^n$ - doubleton
Input: $g : \mathbb{R}^n \to \mathbb{R}^m$ - smooth function
Output: Bound for $g(x + C[r_0] + B[r])$

// enclose set as interval vector
$[X] \leftarrow [x + C[r_0] + B[r]]$
// enclose derivative as interval matrix
$[M] \leftarrow [Dg([X])]$
return $[g([X])] \cap [g(x) + ([M]C)[r_0] + ([M]B)[r]]$

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: $x + C[r_0] + B[r] \subset \mathbb{R}^n$ - doubleton
Input: $g : \mathbb{R}^n \to \mathbb{R}^m$ - smooth function
Output: Bound for $g(x + C[r_0] + B[r])$

// enclose set as interval vector
$[X] \leftarrow [x + C[r_0] + B[r]]$
// enclose derivative as interval matrix
$[M] \leftarrow [Dg([X])]$
return $[g([X])] \cap [g(x) + ([M]C)[r_0] + ([M]B)[r]]$

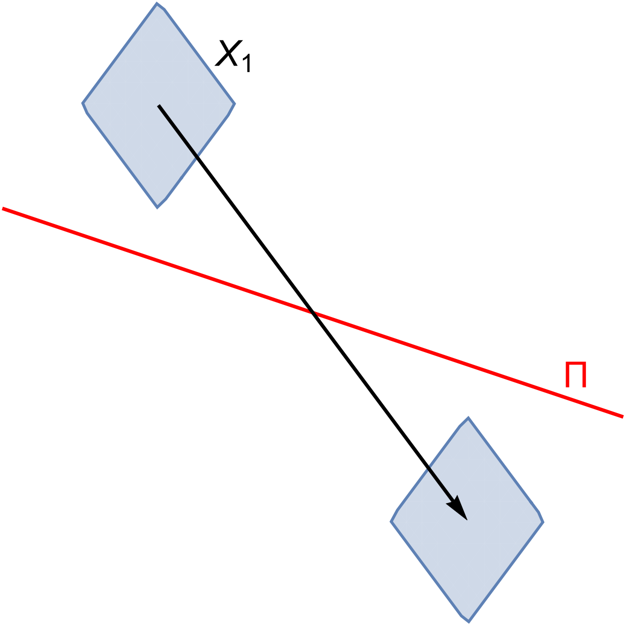
Algorithm: COMPUTEPOINCAREMAP

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 $Q(P([X]) - x_0)$

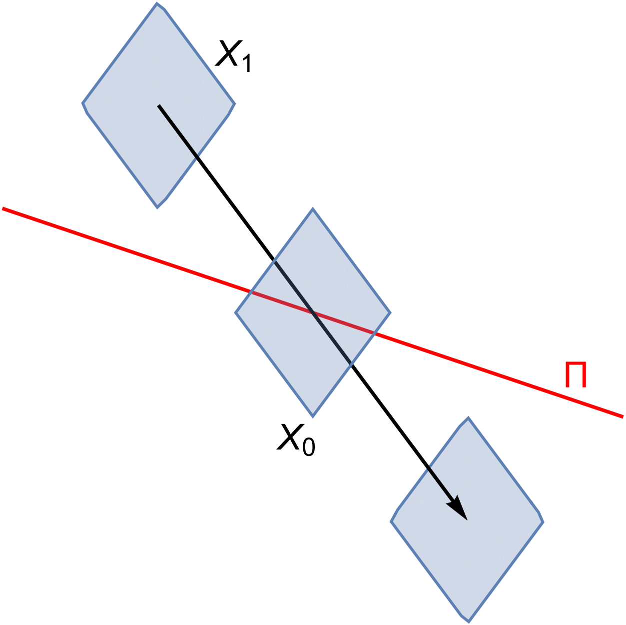
$$
\begin{array}{lcl}
t_0 & \leftarrow & (t_1 + t_2)/2 \\
[\Delta t] & \leftarrow & [t_1, t_2] - t_0 \\
[X_0] & \leftarrow & \texttt{RepresentableSet} \text{ that encloses } \varphi(t_0 - t_1, [X_1]) \\
[Y_0] & \leftarrow & \texttt{affineTransform}([X_0], Q, x_0) \\
[Y] & \leftarrow & \texttt{eval}([X_0], Q \circ f) \cdot [\Delta t] \\
[E] & \leftarrow & \texttt{eval}([X_1], \varphi([0, t_2 - t_1], \cdot)) \\
[\Delta Y] & \leftarrow & \frac{1}{2} Q \cdot Df([E]) \cdot f([E]) \cdot [\Delta t^2] \\
[Z] & \leftarrow & ([Y_0] + [Y] + [\Delta Y]) \cap Q([E] - x_0)
\end{array}
$$

return $[Z]$

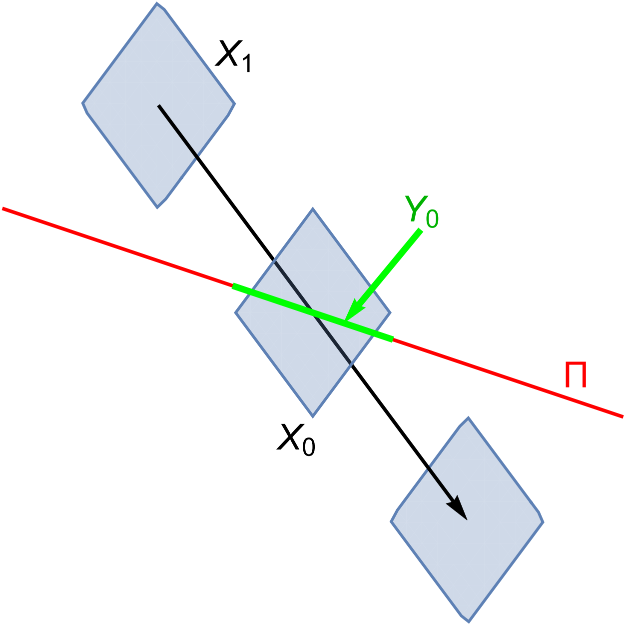
Enclosing Poincaré maps - geometry of the algorithm



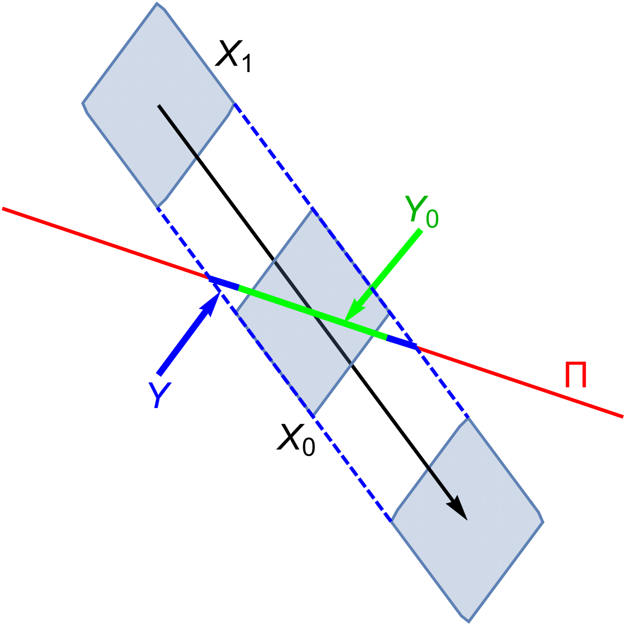
Enclosing Poincaré maps - geometry of the algorithm



Enclosing Poincaré maps - geometry of the algorithm



Enclosing Poincaré maps - geometry of the algorithm



Minimize crossing time:

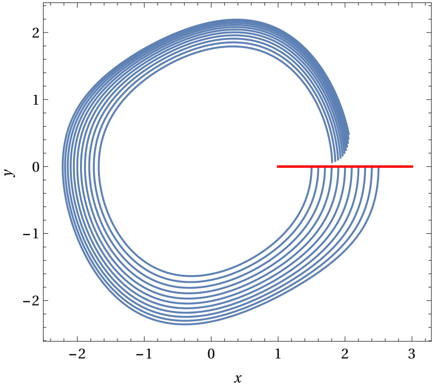
$$Y := \mathtt{eval}(X_0, Q \circ f) \cdot \color{red}{\Delta t}$$
$$\Delta Y := \frac{1}{2} Q \cdot Df(E) \cdot f(E) \cdot \color{red}{\Delta t^2}$$

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

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:

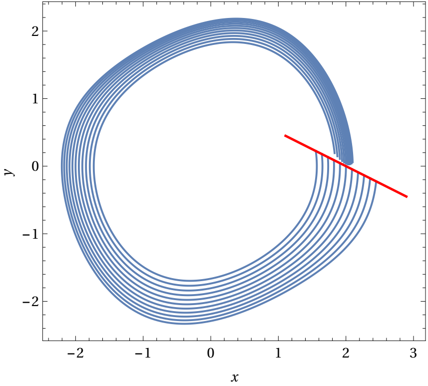
$$Y := \texttt{eval}(X_0, Q \circ f) \cdot {\color{red}\Delta t}$$
$$\Delta Y := \frac{1}{2} Q \cdot Df(E) \cdot f(E) \cdot {\color{red}\Delta t^2}$$

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

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

$$\Downarrow$$

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

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

$$\Downarrow$$

$\nabla\alpha(x)$ 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
$$\Downarrow$$
$\frac{\partial t_\Pi}{\partial v}(x) = 0$ 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

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

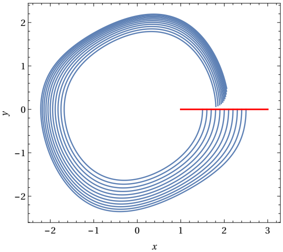
$$\Downarrow$$
$\nabla\alpha(x)$ and $t_\Pi(x)$ are collinear
$$\Downarrow$$
$\frac{\partial t_\Pi}{\partial v}(x) = 0$ for $v \in T_x\Pi$

Example: van der Pol equation

Equation:

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

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



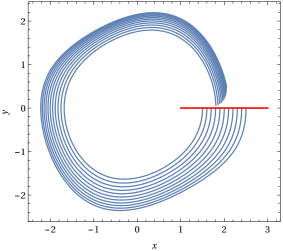
init diam	crossing time	x_0	x_1
1e-10	3.61e-11	[-3.61e-11, 3.61e-11]	[-2.83e-11, 2.83e-11]
1e-09	3.6e-10	[-3.6e-10, 3.6e-10]	[-2.83e-10, 2.83e-10]
1e-08	3.6e-09	[-3.6e-09, 3.6e-09]	[-2.83e-09, 2.83e-09]
1e-07	3.6e-08	[-3.6e-08, 3.6e-08]	[-2.83e-08, 2.83e-08]
1e-06	3.6e-07	[-3.6e-07, 3.6e-07]	[-2.83e-07, 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)

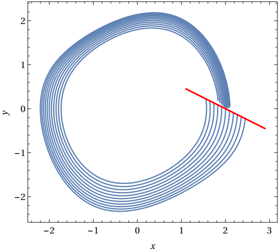


init diam	crossing time	x_0	x_1
1e-10	3.61e-11	[-3.61e-11, 3.61e-11]	[-2.83e-11, 2.83e-11]
1e-09	3.6e-10	[-3.6e-10, 3.6e-10]	[-2.83e-10, 2.83e-10]
1e-08	3.6e-09	[-3.6e-09, 3.6e-09]	[-2.83e-09, 2.83e-09]
1e-07	3.6e-08	[-3.6e-08, 3.6e-08]	[-2.83e-08, 2.83e-08]
1e-06	3.6e-07	[-3.6e-07, 3.6e-07]	[-2.83e-07, 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]

Equation:

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

The section: minimizes crossing time



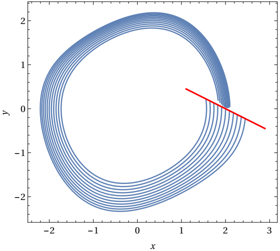
init diam	crossing time	x_0	x_1
1e-10	3.46e-14	[-2.91e-14, 2.93e-14]	[-2.83e-11, 2.83e-11]
1e-09	3.46e-14	[-2.91e-14, 2.93e-14]	[-2.83e-10, 2.83e-10]
1e-08	3.55e-14	[-2.94e-14, 2.95e-14]	[-2.83e-09, 2.83e-09]
1e-07	6.48e-14	[-5.56e-14, 5.58e-14]	[-2.83e-08, 2.83e-08]
1e-06	2.99e-12	[-2.67e-12, 2.67e-12]	[-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]

CAPD library

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 **A**ssisted **P**roofs in **D**ynamics

1992 Marian Mrozek

First version of rigorous ODE solver.

1995 M.Mrozek, K.Mischaikow

Encyclopedia Britanica:

paper on chaos in the Lorenz equations amog 4 best results in mathematics in 1995

2001

CAPD publically available at **http://capd.ii.uj.edu.pl**

1992-2016

continuously developed at the Jagiellonian University

1992 Marian Mrozek

First version of rigorous ODE solver.

1995 M.Mrozek, K.Mischaikow

Encyclopedia Britanica:

paper on chaos in the Lorenz equations amog 4 best results in mathematics in 1995

2001

CAPD publically available at **http://capd.ii.uj.edu.pl**

1992-2016

continuously developed at the Jagiellonian University

1992 Marian Mrozek

First version of rigorous ODE solver.

1995 M.Mrozek, K.Mischaikow

Encyclopedia Britanica:

paper on chaos in the Lorenz equations amog 4 best results in mathematics in 1995

2001

CAPD publically available at **http://capd.ii.uj.edu.pl**

1992-2016

continuously developed at the Jagiellonian University

The CAPD 4.0 in 2016:

- core CAPD: (Multiprecision) IA, linear algebra (dense)
- **capdRedHom: (co)-homology software**

Paweł Pilarczyk

Paweł Dłotko

Mateusz Juda

- **capdDynSys: validated numerics for dynamics**

Piotr Zgliczyński

Tomasz Kapela

Daniel Wilczak

The capdDynSys 4.0 in 2016:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ 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 2016:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ 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 2016:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ 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 2016:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ 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 2016:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ 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 2016:

- $\mathcal{C}^0 - \mathcal{C}^1 - \mathcal{C}^r$ 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

Compilation of the library:

- `./configure [options]`
- `make -j`

  (takes some time, more than 120 000 lines of code)

Basic options:

- `--with-mpfr`
- `--with-wx-config` (internal graphics kernel)

Building own programs:

```
g++ main.cpp -o main `capd-config --cflags --libs`
```

Optional scripts:

```
`capd-gui-config --cflags --libs`
`mpcapd-config --cflags --libs`
```

Compilation of the library:

- `./configure [options]`
- `make -j`

  (takes some time, more than 120 000 lines of code)

Basic options:

- `--with-mpfr`
- `--with-wx-config` (internal graphics kernel)

Building own programs:

```
g++ main.cpp -o main `capd-config --cflags --libs`
```

Optional scripts:

```
`capd-gui-config --cflags --libs`
`mpcapd-config --cflags --libs`
```

Compilation of the library:

- `./configure [options]`
- `make -j`

  (takes some time, more than 120 000 lines of code)

Basic options:

- `--with-mpfr`
- `--with-wx-config` (internal graphics kernel)

Building own programs:

```
g++ main.cpp -o main `capd-config --cflags --libs`
```

Optional scripts:

```
`capd-gui-config --cflags --libs`
`mpcapd-config --cflags --libs`
```

Compilation of the library:

- `./configure [options]`
- `make -j`

  (takes some time, more than 120 000 lines of code)

Basic options:

- `--with-mpfr`
- `--with-wx-config` (internal graphics kernel)

Building own programs:

```
g++ main.cpp -o main `capd-config --cflags --libs`
```

Optional scripts:

```
`capd-gui-config --cflags --libs`
`mpcapd-config --cflags --libs`
```

Main header files

```
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h" // for multiprecision
#include "capd/krak/krak.h" // for graphics kernel
```

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
  - **D**Vector, **LD**Vector, **I**Vector, **Mp**Vector, **MpI**Vector
  - **[Prefix]**Matrix
  - **[Prefix]**Hessian
  - **[Prefix]**Jet
- **Automatic differentiation:**
  - **[Prefix]**Map
- **ODE solvers:**
  - **[Prefix]**OdeSolver, **[Prefix]**CnOdeSolver
  - **[Prefix]**TimeMap, **[Prefix]**CnTimeMap
  - **[Prefix]**PoincareMap, **[Prefix]**CnPoincareMap

Main header files

```
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h" // for multiprecision
#include "capd/krak/krak.h" // for graphics kernel
```

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
  - **D**Vector, **LD**Vector, **I**Vector, **Mp**Vector, **MpI**Vector
  - **[Prefix]**Matrix
  - **[Prefix]**Hessian
  - **[Prefix]**Jet
- **Automatic differentiation:**
  - **[Prefix]**Map
- **ODE solvers:**
  - **[Prefix]**OdeSolver, **[Prefix]**CnOdeSolver
  - **[Prefix]**TimeMap, **[Prefix]**CnTimeMap
  - **[Prefix]**PoincareMap, **[Prefix]**CnPoincareMap

Main header files

```
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h" // for multiprecision
#include "capd/krak/krak.h" // for graphics kernel
```

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
  - **D**Vector, **LD**Vector, **I**Vector, **Mp**Vector, **MpI**Vector
  - **[Prefix]**Matrix
  - **[Prefix]**Hessian
  - **[Prefix]**Jet
- **Automatic differentiation:**
  - **[Prefix]**Map
- **ODE solvers:**
  - **[Prefix]**OdeSolver, **[Prefix]**CnOdeSolver
  - **[Prefix]**TimeMap, **[Prefix]**CnTimeMap
  - **[Prefix]**PoincareMap, **[Prefix]**CnPoincareMap

Main header files

```
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h" // for multiprecision
#include "capd/krak/krak.h" // for graphics kernel
```

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
  - **D**Vector, **LD**Vector, **I**Vector, **Mp**Vector, **MpI**Vector
  - **[Prefix]**Matrix
  - **[Prefix]**Hessian
  - **[Prefix]**Jet
- **Automatic differentiation:**
  - **[Prefix]**Map
- **ODE solvers:**
  - **[Prefix]**OdeSolver, **[Prefix]**CnOdeSolver
  - **[Prefix]**TimeMap, **[Prefix]**CnTimeMap
  - **[Prefix]**PoincareMap, **[Prefix]**CnPoincareMap

Main header files

```
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h" // for multiprecision
#include "capd/krak/krak.h" // for graphics kernel
```

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
  - **D**Vector, **LD**Vector, **I**Vector, **Mp**Vector, **MpI**Vector
  - **[Prefix]**Matrix
  - **[Prefix]**Hessian
  - **[Prefix]**Jet
- **Automatic differentiation:**
  - **[Prefix]**Map
- **ODE solvers:**
  - **[Prefix]**OdeSolver, **[Prefix]**CnOdeSolver
  - **[Prefix]**TimeMap, **[Prefix]**CnTimeMap
  - **[Prefix]**PoincareMap, **[Prefix]**CnPoincareMap

Main header files

```
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h" // for multiprecision
#include "capd/krak/krak.h" // for graphics kernel
```

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
  - **D**Vector, **LD**Vector, **I**Vector, **Mp**Vector, **MpI**Vector
  - **[Prefix]**Matrix
  - **[Prefix]**Hessian
  - **[Prefix]**Jet
- **Automatic differentiation:**
  - **[Prefix]**Map
- **ODE solvers:**
  - **[Prefix]**OdeSolver, **[Prefix]**CnOdeSolver
  - **[Prefix]**TimeMap, **[Prefix]**CnTimeMap
  - **[Prefix]**PoincareMap, **[Prefix]**CnPoincareMap

```cpp
// Note: all examples in this tutorial require C++11 support
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
int main(){
  IMap f("par:a;var:x,y;fun:sin(x*cos(y)),y*cos(a*y)-x^2;",3);
  f.setParameter("a",interval(1.,1.01));
  IVector u({1,3});
  cout << "f(u)=" << f(u) << endl;
  cout << "Df(u)=" << f.derivative(u) << endl;
  IJet jet(2,3);   // third order Taylor coefficients, two variables
  f(u,jet);        // compute full Taylor expansion
  cout << jet.toString() << endl;

  IOdeSolver solver(f,20);  // ODE integrator
  ITimeMap tm(solver);      // class for long time integration
  C0HOTripletonSet set(u);  // representation of initial condition
  // integrate until T=2 and print result
  cout << "phi(2,(1,3)) = " << tm(2.,set) << endl;

  ITimeMap::SolutionCurve solution(0.);
  set = C0HOTripletonSet(IVector({1,1}));
  // integrate until T=2 and save entire trajectory
  tm(2.,set,solution);
  cout << "solution(2)=" << solution(2) << endl;
  cout << "solution(0.9,1.1)=" << solution(interval(.9,1.1)) << endl;
}
```

```
// Note: all examples in this tutorial require C++11 support
#include <iostream>
#include "capd/capdlib.h"
#include "capd/mpcapdlib.h"
using namespace capd;
using namespace std;
int main(){
  cout.precision(60);
  MpFloat::setDefaultPrecision(200);
  MpIMap lorenz("var:x,y,z;fun:10*(y-x),x*(28-z)-y,x*y-8*z/3;");

  MpIOdeSolver solver(lorenz,40);   // ODE integrator
  MpITimeMap tm(solver);            // class for long time integration
  // initial condition
  MpIVector u({MpInterval(1.),MpInterval(3.),MpInterval(10.)});
  // tripleton representation of initial condition
  MpC0TripletonSet set(u);
  // integrate until T=2 and print result
  cout << "phi(2,(1,3,10)) = " << tm(MpInterval(2.),set) << endl;
}
/* Output (reformatted to fit presentation window)
 phi(2,(1,3,10)) = {
 [-2.173368855117490127189994870441944795784915747510543760726653 ,
  -2.17336885511749012718999487044194479578491574751054376063012 ],
 [-1.81501617053155848885754667427341477756244393385646228596035 7 ,
  -1.81501617053155848885754667427341477756244393385646228594609 2 ],
 [2.04129457433159136349537343394345779855485364177570327972290e1 ,
  2.04129457433159136349537343394345779855485364177570327973768e1 ]}*/
```

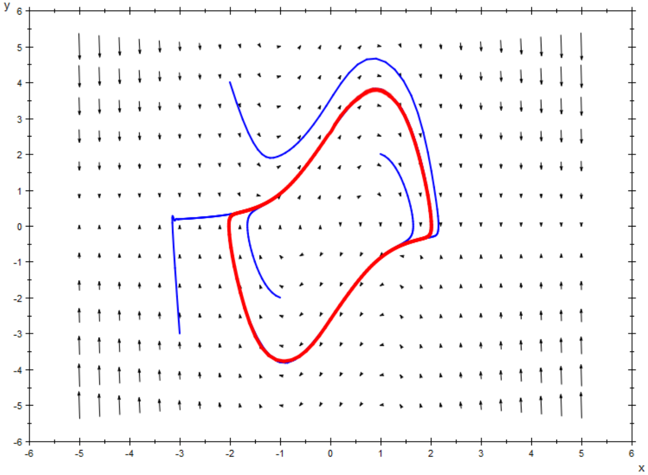
Case study

- (symmetric) periodic orbits
- existence of an attractor

Example (Toy example: van der Pol oscillator)

$$x'' = \mu(1 - x^2) * x' - x$$

Goal: van der Pol oscillator has a periodic solution for $\mu = 1$.



Picture taken from Wikipedia

Van der Pol oscillator

Settings:

$$\begin{array}{lcl} \Pi = \{(x,0) : x \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi \to \Pi & - & \text{Poincaré map} \end{array}$$

Methodology:

- $x_0 \in \Pi$ – an approximate periodic point for P
  (skip coordinate $x' = 0$)
- r – an interval centred at zero
- Check that

$$P(x_0 + r) \subset x_0 + r$$

Result: Success with `r=interval(-1,1)*1e-5`.

Hyperlink: complete C++-11 source code

Van der Pol oscillator

Settings:

$$\Pi = \{(x,0) : x \in \mathbb{R}\} \quad - \quad \text{Poincaré section}$$
$$P : \Pi \to \Pi \quad - \quad \text{Poincaré map}$$

Methodology:

- $x_0 \in \Pi$ – an approximate periodic point for P
  (skip coordinate $x' = 0$)
- r – an interval centred at zero
- Check that

$$P(x_0 + r) \subset x_0 + r$$

Result: Success with `r=interval(-1,1)*1e-5`.

Hyperlink: complete C++-11 source code

Van der Pol oscillator

Settings:

$$
\begin{array}{lcl}
\Pi = \{(x, 0) : x \in \mathbb{R}\} & - & \text{Poincaré section} \\
P : \Pi \to \Pi & - & \text{Poincaré map}
\end{array}
$$

Methodology:

- $x_0 \in \Pi$ – an approximate periodic point for P
  (skip coordinate $x' = 0$)
- r – an interval centred at zero
- Check that

$$P(x_0 + r) \subset x_0 + r$$

Result: Success with `r=interval(-1,1)*1e-5`.

Hyperlink: complete C++-11 source code

Van der Pol oscillator

Settings:

$$\Pi = \{(x,0) : x \in \mathbb{R}\} \quad - \quad \text{Poincaré section}$$
$$P : \Pi \to \Pi \quad - \quad \text{Poincaré map}$$

Methodology:

- $x_0 \in \Pi$ – an approximate periodic point for P
  (skip coordinate $x' = 0$)
- r – an interval centred at zero
- Check that
$$P(x_0 + r) \subset x_0 + r$$

Result: Success with `r=interval(-1,1)*1e-5`.

Hyperlink: complete C++-11 source code

Van der Pol oscillator

Settings:

$$\begin{array}{lcl} \Pi = \{(x,0) : x \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi \to \Pi & - & \text{Poincaré map} \end{array}$$

Methodology:

- $x_0 \in \Pi$ – an approximate periodic point for P
  (skip coordinate $x' = 0$)
- r – an interval centred at zero
- Check that

$$P(x_0 + r) \subset x_0 + r$$

Result: Success with `r=interval(-1,1)*1e-5`.

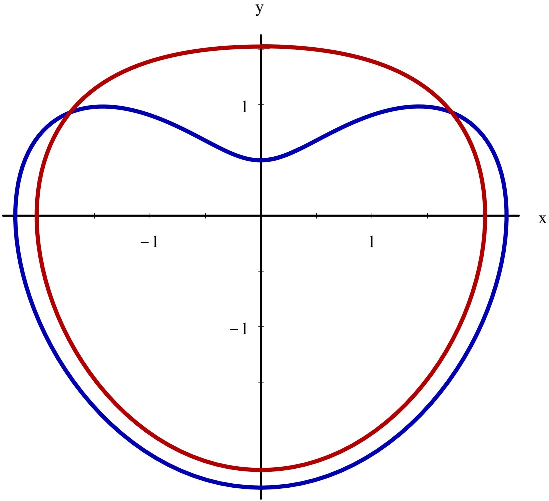
Hyperlink: complete C++-11 source code

```cpp
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
int main(){
  // Instance of the vector field, ODE solver,
  // Poincare section (y=0 is index=1 of 2 variables)
  // and the crossing direction (y changes sign from Plus to Minus)
  IMap vf("var:x,y;fun:y,(1-x^2)*y-x;");
  IOdeSolver solver(vf, 20);
  ICoordinateSection section(2, 1);
  IPoincareMap pm(solver, section, poincare::PlusMinus);

  // Take a ball centred at approximate periodic point
  IVector x({2.0086198608748433 + interval(-1,1)*1e-5,0.});
  C0HOTripletonSet s(x);
  // Call routine that computes rigorously Poincare map
  IVector y = pm(s);
  // check inclusion and print output
  cout.precision(17);
  cout << "y=" << y[0] << "\nx=" << x[0] << endl;
  cout << boolalpha << "inclusion? = " << subset(y[0],x[0]);
  return 0;
}
/* Output:
y=[2.0086198481018909, 2.0086198740315355]
x=[2.008609860874843З, 2.0086298608748434]
inclusion? = true */
```

Example (Toy example: Michelson system)

$$x' = y, \qquad y' = z, \qquad z' = 1 - y - x^2/2$$

Goal:
the system has at least two symmetric periodic solutions.



Michelson system

Settings:

$$\Pi = \{(x,y,0) : x,y \in \mathbb{R}\} \quad - \quad \text{Poincaré section}$$
$$P : \Pi \to \Pi \quad - \quad \text{Poincaré map}$$
$$R(x,y) = (-x,y) \quad - \quad \text{reversing symmetry for } P$$

Methodology:

- $R \circ P^n \circ R \circ P^n = \mathrm{Id}$ (provided left side is defined)
- If $\pi_x P^n(0, y_0) = 0$ then y_0 is a symmetric periodic point
- $(0, y) \in \Pi$ – an approximate periodic point for P^n
- $r > 0$ – constant
- Check that P^n is defined on $(0, [y-r, y+r])$ and

$$P^n(0, y-r) \cdot P^n(0, y+r) < 0$$

Result: Success with `r=interval(-1,1)*1e-13`.

Hyperlink: complete C++-11 source code

Settings:

$$\begin{array}{lcl} \Pi = \{(x,y,0) : x,y \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi \to \Pi & - & \text{Poincaré map} \\ R(x,y) = (-x,y) & - & \text{reversing symmetry for } P \end{array}$$

Methodology:

- $R \circ P^n \circ R \circ P^n = \mathrm{Id}$ (provided left side is defined)
- If $\pi_x P^n(0, y_0) = 0$ then y_0 is a symmetric periodic point
- $(0, y) \in \Pi$ – an approximate periodic point for P^n
- $r > 0$ – constant
- Check that P^n is defined on $(0, [y - r, y + r])$ and

$$P^n(0, y - r) \cdot P^n(0, y + r) < 0$$

Result: Success with `r=interval(-1,1)*1e-13`.

Hyperlink: complete C++-11 source code

Michelson system

Settings:

$$
\begin{array}{lcl}
\Pi = \{(x,y,0) : x,y \in \mathbb{R}\} & - & \text{Poincaré section} \\
P : \Pi \to \Pi & - & \text{Poincaré map} \\
R(x,y) = (-x,y) & - & \text{reversing symmetry for } P
\end{array}
$$

Methodology:

- $R \circ P^n \circ R \circ P^n = \mathrm{Id}$ (provided left side is defined)
- If $\pi_x P^n(0, y_0) = 0$ then y_0 is a symmetric periodic point
- $(0, y) \in \Pi$ – an approximate periodic point for P^n
- $r > 0$ – constant
- Check that P^n is defined on $(0, [y - r, y + r])$ and

$$P^n(0, y - r) \cdot P^n(0, y + r) < 0$$

Result: Success with `r=interval(-1,1)*1e-13`.

Hyperlink: complete C++-11 source code

Michelson system

Settings:

$$\begin{array}{lcl} \Pi = \{(x,y,0) : x,y \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi \to \Pi & - & \text{Poincaré map} \\ R(x,y) = (-x,y) & - & \text{reversing symmetry for } P \end{array}$$

Methodology:

- $R \circ P^n \circ R \circ P^n = \mathrm{Id}$ (provided left side is defined)
- If $\pi_x P^n(0, y_0) = 0$ then y_0 is a symmetric periodic point
- $(0, y) \in \Pi$ – an approximate periodic point for P^n
- $r > 0$ – constant
- Check that P^n is defined on $(0, [y - r, y + r])$ and

$$P^n(0, y - r) \cdot P^n(0, y + r) < 0$$

Result: Success with `r=interval(-1,1)*1e-13`.

Hyperlink: complete C++-11 source code

Michelson system

Settings:

$$
\begin{array}{lcl}
\Pi = \{(x, y, 0) : x, y \in \mathbb{R}\} & - & \text{Poincaré section} \\
P : \Pi \to \Pi & - & \text{Poincaré map} \\
R(x, y) = (-x, y) & - & \text{reversing symmetry for } P
\end{array}
$$

Methodology:

- $R \circ P^n \circ R \circ P^n = \mathrm{Id}$ (provided left side is defined)
- If $\pi_x P^n(0, y_0) = 0$ then y_0 is a symmetric periodic point
- $(0, y) \in \Pi$ – an approximate periodic point for P^n
- $r > 0$ – constant
- Check that P^n is defined on $(0, [y - r, y + r])$ and

$$P^n(0, y - r) \cdot P^n(0, y + r) < 0$$

Result: Success with `r=interval(-1,1)*1e-13`.

Hyperlink: complete C++-11 source code

Michelson system

Settings:

$\Pi = \{(x, y, 0) : x, y \in \mathbb{R}\}$	–	Poincaré section
$P : \Pi \to \Pi$	–	Poincaré map
$R(x, y) = (-x, y)$	–	reversing symmetry for P

Methodology:

- $R \circ P^n \circ R \circ P^n = \mathrm{Id}$ (provided left side is defined)
- If $\pi_x P^n(0, y_0) = 0$ then y_0 is a symmetric periodic point
- $(0, y) \in \Pi$ – an approximate periodic point for P^n
- $r > 0$ – constant
- Check that P^n is defined on $(0, [y - r, y + r])$ and

$$P^n(0, y - r) \cdot P^n(0, y + r) < 0$$

Result: Success with `r=interval(-1,1)*1e-13`.

Hyperlink: complete C++-11 source code

```cpp
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;

void validateSymPO(interval y, int iteration){
  IMap vf("var:x,y,z;fun:y,z,1-y-0.5*x^2;"); // vector field
  IOdeSolver solver(vf, 20);                 // 20th order solver
  ICoordinateSection section(3, 2);          // section z=0
  IPoincareMap pm(solver, section);

  // Check that the Poincare map is defined on (0,y)
  C0HOTripletonSet s(IVector({0.,y,0.}));
  std::cout << "P(0,y)=" << pm(s,iteration) << std::endl;

  // Compute P(0,left(y)) and P(0,right(y))
  C0HOTripletonSet s1(IVector({0.,y.left(),0.}));
  C0HOTripletonSet s2(IVector({0.,y.right(),0.}));
  IVector r1 = pm(s1,iteration);
  IVector r2 = pm(s2,iteration);
  // Check that x-coordinate changes the sign
  std::cout << "check inclusion? " << ( r1[0]*r2[0]<0 ) << std::endl;
}
int main(){
  // Call the algorithm with two approximate periodic points
  validateSymPO(1.5259617305036892  + interval(-1,1)*1e-15, 1);
  validateSymPO(0.50002564853520548 + interval(-1,1)*1e-13, 2);
  return 0;
}
```

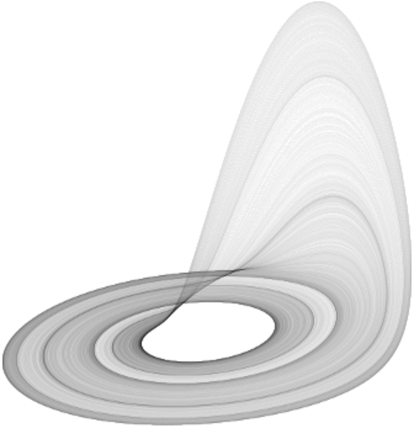
Example: attractor in the Rössler system

Example (Rössler system)

$$x' = -(y+z), \qquad y' = x + 0.2y, \qquad z' = 0.2 + z(x - 5.7)$$

Goal:

there is a compact, connected invariant set which has at least one periodic solution.

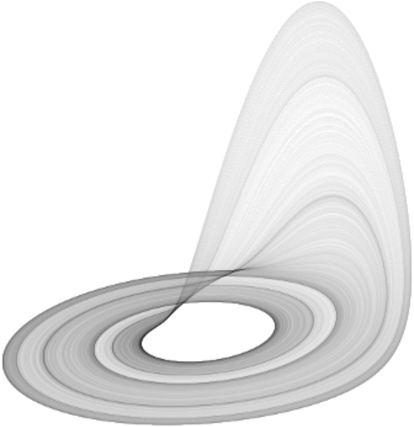


Example: attractor in the Rössler system

Example (Rössler system)

$$x' = -(y+z), \qquad y' = x + 0.2y, \qquad z' = 0.2 + z(x - 5.7)$$

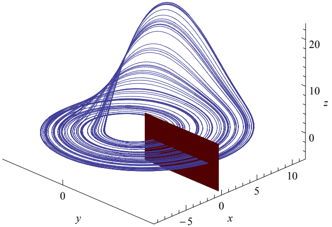
Goal:
there is a compact, connected invariant set which has at least one periodic solution.



Example: attractor in the Rössler system

Settings:

$$\Pi = \{(0, y, z) : y, z \in \mathbb{R}, x' > 0\} \quad - \quad \text{Poincaré section}$$
$$P : \Pi \to \Pi \quad - \quad \text{Poincaré map}$$



Methodology: Show that there is a rectangle

$$W = [y_1, y_2] \times [z_1, z_2]$$

such that

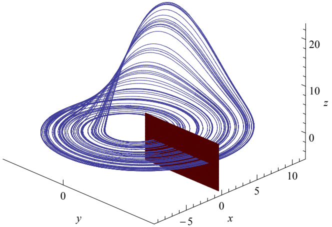
$$P(W) \subset W.$$

Then $\mathcal{A} := \bigcap_{n>0} P^n(W)$ is a compact, connected invariant set.

Example: attractor in the Rössler system

Settings:

$$\Pi = \{(0, y, z) : y, z \in \mathbb{R}, x' > 0\} \quad - \quad \text{Poincaré section}$$
$$P : \Pi \to \Pi \quad - \quad \text{Poincaré map}$$



Methodology: Show that there is a rectangle

$$W = [y_1, y_2] \times [z_1, z_2]$$

such that

$$P(W) \subset W.$$

Then $\mathcal{A} := \bigcap_{n>0} P^n(W)$ is a compact, connected invariant set.

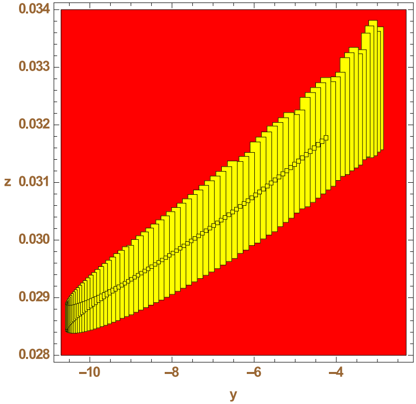
Example: attractor in the Rössler system

Data (from simulation):

$$W = [-10.7, -2.3] \times [0.028, 0.034]$$

Computations:

- subdivide $W = \bigcup_{i=1}^{200} W_i$
- check that $P(W_i) \subset W$ for $i = 1, \ldots, 200$



```cpp
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;

int main(){
  IMap vf("var:x,y,z;fun:-(y+z),x+0.2*y,0.2+z*(x-5.7);");
  IOdeSolver solver(vf, 20);
  ICoordinateSection section(3, 0); // section x=0, x'>0
  IPoincareMap pm(solver, section, poincare::MinusPlus);

  // Coordinates of the trapping region
  const double B = 0.028, T = 0.034, L = -10.7, R = -2.3;

  // Subdivide uniformly onto 200 rectangles
  const int N = 160;
  bool result = true;
  interval p = (interval(R) - interval(L)) / N;
  for (int i = 0; i < N and result; ++i) {
    IVector x ({0., L + interval(i,i+1)*p, interval(B, T)});
    C0HOTripletonSet s(x);
    IVector u = pm(s);
    result = result and u[2]>B and u[2]<T and u[1]>L and u[1]<R;
    if(!result)
      std::cout << "Inclusion not satisfied:\n" << u << std::endl;
  }
  std::cout << "Existence of attractor: " << result << std::endl;
  return 0;
}
```

Case study

existence of chaos

Theorem (Zgliczyński, Nonlinearity 1997)

The Rössler system

$$x' = -(y+z), \quad y' = x + 0.2y, \quad z' = 0.2 + z(x - 5.7)$$

is chaotic.

Here we need "good theorem" of the form:

finite number of inequalities

$$\Downarrow$$

chaos

P. Zgliczyński, Computer assisted proof of chaos in the Hénon map and in the Rössler equations, Nonlinearity, 1997, Vol. 10, No. 1, 243–252

Theorem (Zgliczyński, Nonlinearity 1997)

The Rössler system

$$x' = -(y+z), \quad y' = x + 0.2y, \quad z' = 0.2 + z(x - 5.7)$$

is chaotic.

Here we need "good theorem" of the form:

finite number of inequalities

$$\Downarrow$$

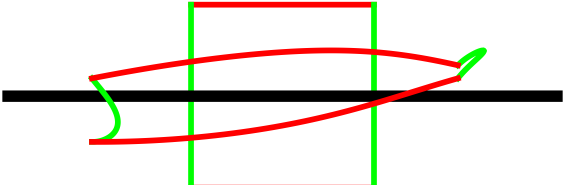
chaos

P. Zgliczyński, Computer assisted proof of chaos in the Hénon map and in the Rössler equations, Nonlinearity, 1997, Vol. 10, No. 1, 243–252

Definition

Let
$$N = [l_N, r_N] \times [b_N, t_N], \quad M = [l_M, r_M] \times [b_M, t_M]$$
and let $f : N \to \mathbb{R}^2$ be continuous. We say that N f-covers M, denoted by
$$N \overset{f}{\Longrightarrow} M,$$
if

- $P(N) \subset \mathbb{R} \times (b_M, t_M)$ and
- either $f(l_N \times [b_N, t_N]) < l_M$ and $f(r_N \times [b_N, t_N]) > r_M$
- or $f(l_N \times [b_N, t_N]) > r_M$ and $f(r_N \times [b_N, t_N]) < l_M$



Definition

Let

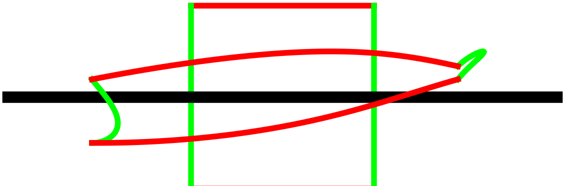
$$N = [l_N, r_N] \times [b_N, t_N], \quad M = [l_M, r_M] \times [b_M, t_M]$$

and let $f : N \to \mathbb{R}^2$ be continuous. We say that N f-covers M, denoted by

$$N \stackrel{f}{\Longrightarrow} M,$$

if

- $P(N) \subset \mathbb{R} \times (b_M, t_M)$ and
- either $f(l_N \times [b_N, t_N]) < l_M$ and $f(r_N \times [b_N, t_N]) > r_M$
- or $f(l_N \times [b_N, t_N]) > r_M$ and $f(r_N \times [b_N, t_N]) < l_M$



Definition

Let

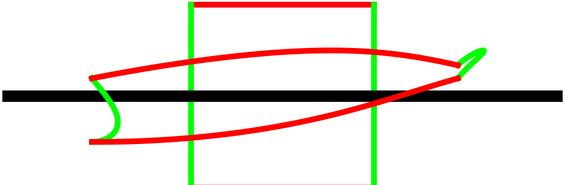
$$N = [l_N, r_N] \times [b_N, t_N], \quad M = [l_M, r_M] \times [b_M, t_M]$$

and let $f : N \to \mathbb{R}^2$ be continuous. We say that N f-covers M, denoted by

$$N \overset{f}{\Longrightarrow} M,$$

if

- $P(N) \subset \mathbb{R} \times (b_M, t_M)$ and
- either $f(l_N \times [b_N, t_N]) < l_M$ and $f(r_N \times [b_N, t_N]) > r_M$
- or $f(l_N \times [b_N, t_N]) > r_M$ and $f(r_N \times [b_N, t_N]) < l_M$

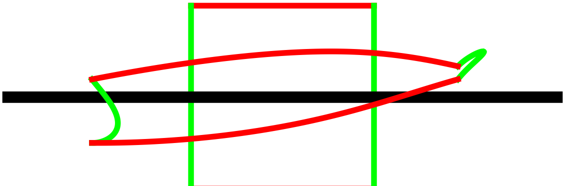


Definition

Let
$$N = [l_N, r_N] \times [b_N, t_N], \quad M = [l_M, r_M] \times [b_M, t_M]$$
and let $f : N \to \mathbb{R}^2$ be continuous. We say that N f-covers M, denoted by
$$N \stackrel{f}{\Longrightarrow} M,$$

if

- $P(N) \subset \mathbb{R} \times (b_M, t_M)$ and
- either $f(l_N \times [b_N, t_N]) < l_M$ and $f(r_N \times [b_N, t_N]) > r_M$
- or $f(l_N \times [b_N, t_N]) > r_M$ and $f(r_N \times [b_N, t_N]) < l_M$



Definition

Let

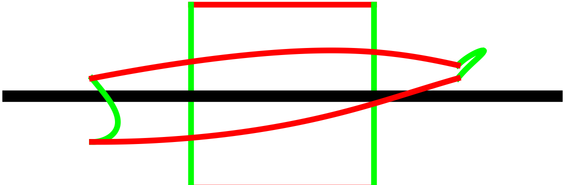
$$N = [l_N, r_N] \times [b_N, t_N], \quad M = [l_M, r_M] \times [b_M, t_M]$$

and let $f : N \to \mathbb{R}^2$ be continuous. We say that N f-covers M, denoted by

$$N \stackrel{f}{\Longrightarrow} M,$$

if

- $P(N) \subset \mathbb{R} \times (b_M, t_M)$ and
- either $f(l_N \times [b_N, t_N]) < l_M$ and $f(r_N \times [b_N, t_N]) > r_M$
- or $f(l_N \times [b_N, t_N]) > r_M$ and $f(r_N \times [b_N, t_N]) < l_M$



Here is a simplified version of theorem by Zgliczyński.

Theorem (Zgliczyński, Nonlinearity 1997)

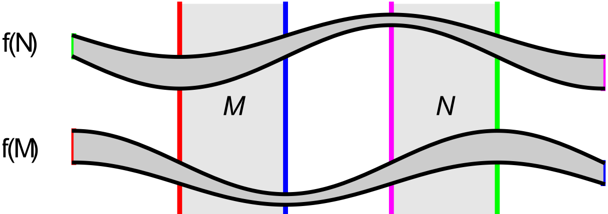
Assume that N and M are disjoint and

$$N \overset{f}{\Longrightarrow} N \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} N.$$

Then for every biinfinite sequence $\{S_i\}_{i\in\mathbb{Z}} \in \{N, M\}^{\mathbb{Z}}$ there is a sequence $\{x_i\}_{i\in\mathbb{Z}} \subset \mathbb{R}^2$ such that for $i \in \mathbb{Z}$ there holds

$$x_i \in S_i, \qquad f(x_i) = x_{i+1}.$$

Periodic $\{S_i\}$ can be realized by periodic $\{x_i\}$ with the same principal period.



Here is a simplified version of theorem by Zgliczyński.

Theorem (Zgliczyński, Nonlinearity 1997)

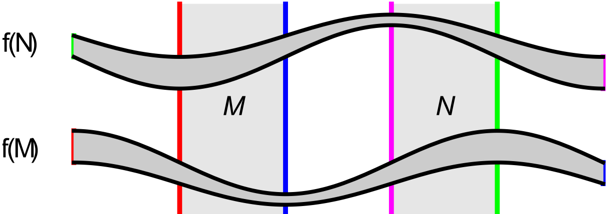
Assume that N and M are disjoint and

$$N \xrightarrow{f} N \xrightarrow{f} M \xrightarrow{f} M \xrightarrow{f} N.$$

Then for every biinfinite sequence $\{S_i\}_{i\in\mathbb{Z}} \in \{N, M\}^{\mathbb{Z}}$ there is a sequence $\{x_i\}_{i\in\mathbb{Z}} \subset \mathbb{R}^2$ such that for $i \in \mathbb{Z}$ there holds

$$x_i \in S_i, \qquad f(x_i) = x_{i+1}.$$

Periodic $\{S_i\}$ can be realized by periodic $\{x_i\}$ with the same principal period.



Here is a simplified version of theorem by Zgliczyński.

Theorem (Zgliczyński, Nonlinearity 1997)

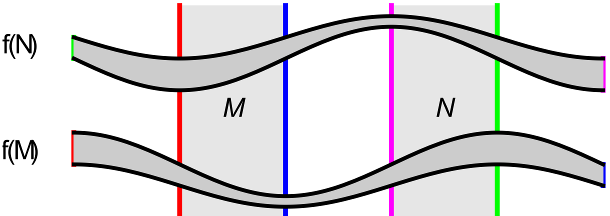
Assume that N and M are disjoint and

$$N \overset{f}{\Longrightarrow} N \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} N.$$

Then for every biinfinite sequence $\{S_i\}_{i\in\mathbb{Z}} \in \{N, M\}^{\mathbb{Z}}$ there is a sequence $\{x_i\}_{i\in\mathbb{Z}} \subset \mathbb{R}^2$ such that for $i \in \mathbb{Z}$ there holds

$$x_i \in S_i, \qquad f(x_i) = x_{i+1}.$$

Periodic $\{S_i\}$ can be realized by periodic $\{x_i\}$ with the same principal period.



Data (from simulation):

$$
\begin{aligned}
W &= [l_W, r_W] \times Z = [-10.7, -2.3] \times [0.028, 0.034] \\
M &= [l_M, r_M] \times Z = [-8.4, -7.6] \times [0.028, 0.034] \\
N &= [l_N, r_N] \times Z = [-5.7, -4.6] \times [0.028, 0.034].
\end{aligned}
$$

Note: in the last example we checked $P(W) \subset \operatorname{int} W$.
Therefore

$$P^2(W) \subset \operatorname{int} W \subset \mathbb{R} \times (0.028, 0.034)$$

Lemma (computer-assisted)

$$N \overset{P^2}{\Longrightarrow} N \overset{P^2}{\Longrightarrow} M \overset{P^2}{\Longrightarrow} M \overset{P^2}{\Longrightarrow} N$$

Inequalities to check:

$$
\begin{aligned}
\pi_y P^2(l_M \times [0.028, 0.034]) &< l_M \\
\pi_y P^2(r_M \times [0.028, 0.034]) &> r_N \\
\pi_y P^2(l_N \times [0.028, 0.034]) &> r_N \\
\pi_y P^2(r_N \times [0.028, 0.034]) &< l_M
\end{aligned}
$$

Data (from simulation):

$$
\begin{aligned}
W &= [l_W, r_W] \times Z = [-10.7, -2.3] \times [0.028, 0.034] \\
M &= [l_M, r_M] \times Z = [-8.4, -7.6] \times [0.028, 0.034] \\
N &= [l_N, r_N] \times Z = [-5.7, -4.6] \times [0.028, 0.034].
\end{aligned}
$$

Note: in the last example we checked $P(W) \subset \operatorname{int} W$.
Therefore

$$P^2(W) \subset \operatorname{int} W \subset \mathbb{R} \times (0.028, 0.034)$$

Lemma (computer-assisted)

$$N \xrightarrow{P^2} N \xrightarrow{P^2} M \xrightarrow{P^2} M \xrightarrow{P^2} N$$

Inequalities to check:

$$
\begin{aligned}
\pi_y P^2(l_M \times [0.028, 0.034]) &< l_M \\
\pi_y P^2(r_M \times [0.028, 0.034]) &> r_N \\
\pi_y P^2(l_N \times [0.028, 0.034]) &> r_N \\
\pi_y P^2(r_N \times [0.028, 0.034]) &< l_M
\end{aligned}
$$

Data (from simulation):

$$\begin{aligned} W &= [l_W, r_W] \times Z = [-10.7, -2.3] \times [0.028, 0.034] \\ M &= [l_M, r_M] \times Z = [-8.4, -7.6] \times [0.028, 0.034] \\ N &= [l_N, r_N] \times Z = [-5.7, -4.6] \times [0.028, 0.034]. \end{aligned}$$

Note: in the last example we checked $P(W) \subset \operatorname{int} W$.
Therefore

$$P^2(W) \subset \operatorname{int} W \subset \mathbb{R} \times (0.028, 0.034)$$

Lemma (computer-assisted)

$$N \xrightarrow{P^2} N \xrightarrow{P^2} M \xrightarrow{P^2} M \xrightarrow{P^2} N$$

Inequalities to check:

$$\begin{aligned} \pi_y P^2(l_M \times [0.028, 0.034]) &< l_M \\ \pi_y P^2(r_M \times [0.028, 0.034]) &> r_N \\ \pi_y P^2(l_N \times [0.028, 0.034]) &> r_N \\ \pi_y P^2(r_N \times [0.028, 0.034]) &< l_M \end{aligned}$$

Data (from simulation):

$$\begin{aligned} W &= [l_W, r_W] \times Z = [-10.7, -2.3] \times [0.028, 0.034] \\ M &= [l_M, r_M] \times Z = [-8.4, -7.6] \times [0.028, 0.034] \\ N &= [l_N, r_N] \times Z = [-5.7, -4.6] \times [0.028, 0.034]. \end{aligned}$$

Note: in the last example we checked $P(W) \subset \operatorname{int} W$.
Therefore

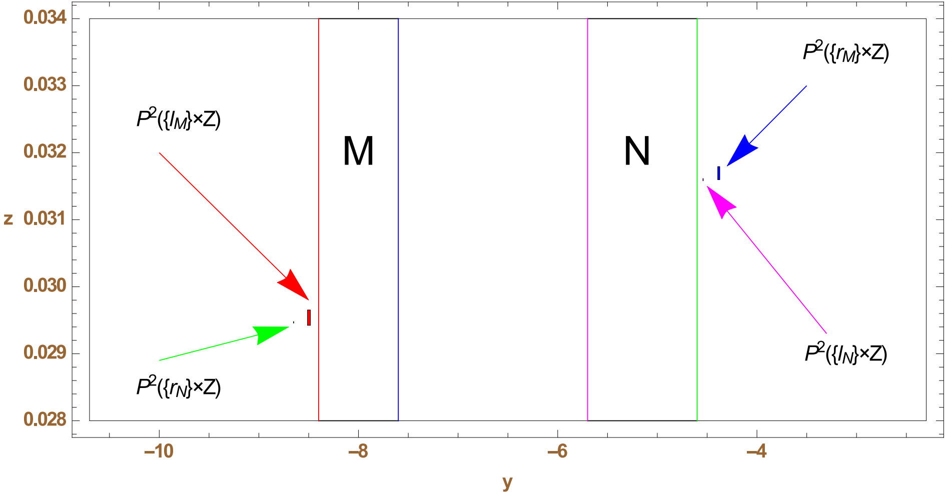
$$P^2(W) \subset \operatorname{int} W \subset \mathbb{R} \times (0.028, 0.034)$$

Lemma (computer-assisted)

$$N \stackrel{P^2}{\Longrightarrow} N \stackrel{P^2}{\Longrightarrow} M \stackrel{P^2}{\Longrightarrow} M \stackrel{P^2}{\Longrightarrow} N$$

Inequalities to check:

$$\begin{aligned} \pi_y P^2(l_M \times [0.028, 0.034]) &< l_M \\ \pi_y P^2(r_M \times [0.028, 0.034]) &> r_N \\ \pi_y P^2(l_N \times [0.028, 0.034]) &> r_N \\ \pi_y P^2(r_N \times [0.028, 0.034]) &< l_M \end{aligned}$$



Rigorous enclosures returned by the routine

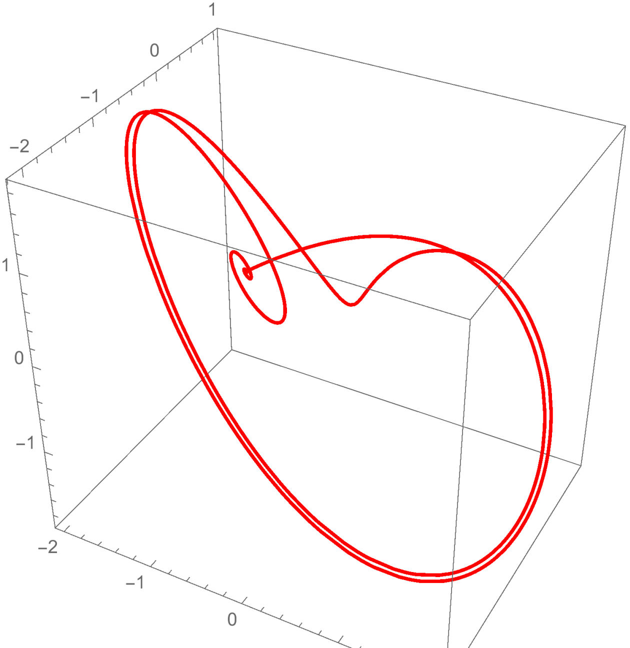
```cpp
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;

int main(){
  IMap vf("var:x,y,z;fun:-(y+z),x+0.2*y,0.2+z*(x-5.7);");
  IOdeSolver solver(vf, 20);
  ICoordinateSection section(3, 0); // section x=0, x'>0
  IPoincareMap pm(solver, section, poincare::MinusPlus);

  // z-coordinate of the trapping region
  interval z(0.028,0.034);
  // Coordinates of M and N
  const double lM=-8.4, rM=-7.6, lN=-5.7, rN=-4.6;

  C0HOTripletonSet LM( IVector({0.,lM,z}) );
  C0HOTripletonSet RM( IVector({0.,rM,z}) );
  C0HOTripletonSet LN( IVector({0.,lN,z}) );
  C0HOTripletonSet RN( IVector({0.,rN,z}) );

  // Inequalities for the covering relations N=>N, N=>M, M=>M, M=>N.
  cout << "P^2(LM) < lM: " << ( pm(LM,2)[1] < lM ) << endl;
  cout << "P^2(RM) > rN: " << ( pm(RM,2)[1] > rN ) << endl;
  cout << "P^2(LN) > rN: " << ( pm(LN,2)[1] > rN ) << endl;
  cout << "P^2(RN) < lM: " << ( pm(RN,2)[1] < lM ) << endl;
  return 0;
}
```

Applications

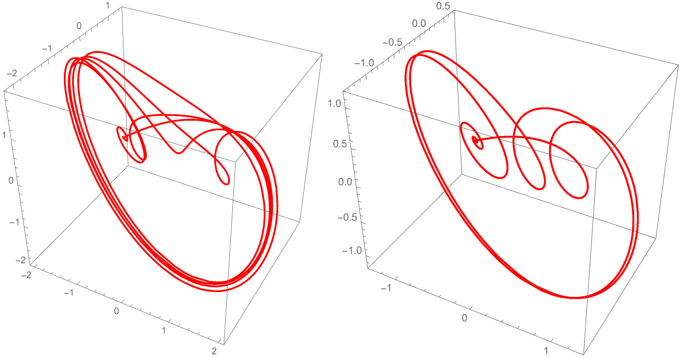
Homoclinic and heteroclinic bifurcations:

- Shilnikov homoclinic orbits
- Bykov heteroclinic cycles

Shilnikov homoclinic orbits



Shilnikov homoclinic orbits

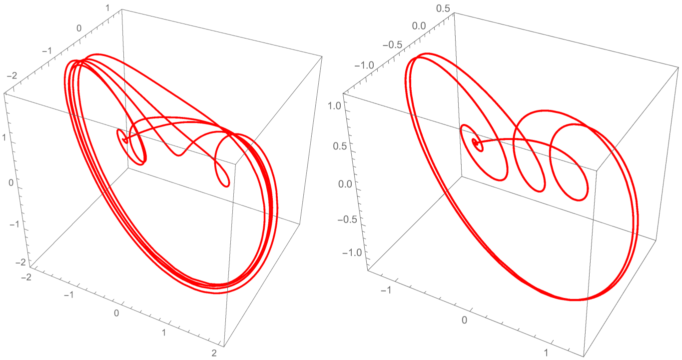


Geometry:

- hyperbolic saddle with 1D unstable(stable) manifold and 2D stable(unstable) manifold with complex eigenvalues
- one branch of 1D unstable manifold $\subset$ 2D stable manifold
- exists for an isolated parameter value (codim 1 bifurcation)

  (destroyed generically by arbitrary small perturbation)

Shilnikov homoclinic orbits

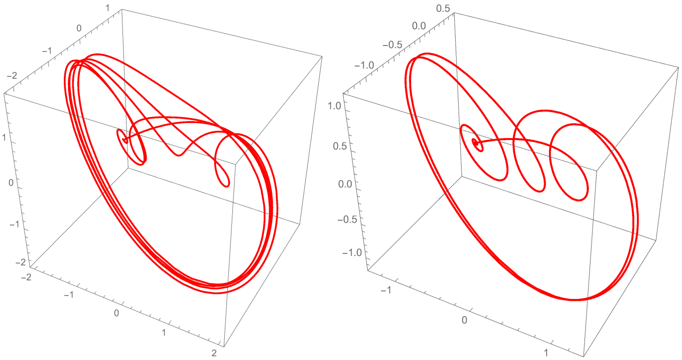


Geometry:

- hyperbolic saddle with 1D unstable(stable) manifold and 2D stable(unstable) manifold with complex eigenvalues
- one branch of 1D unstable manifold $\subset$ 2D stable manifold
- exists for an isolated parameter value (codim 1 bifurcation)

(destroyed generically by arbitrary small perturbation)

Shilnikov homoclinic orbits

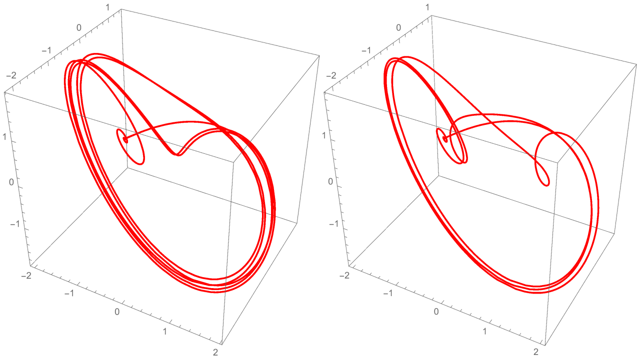


Geometry:

- hyperbolic saddle with 1D unstable(stable) manifold and 2D stable(unstable) manifold with complex eigenvalues
- one branch of 1D unstable manifold $\subset$ 2D stable manifold
- exists for an isolated parameter value (codim 1 bifurcation)

  (destroyed generically by arbitrary small perturbation)

Shilnikov homoclinic orbits



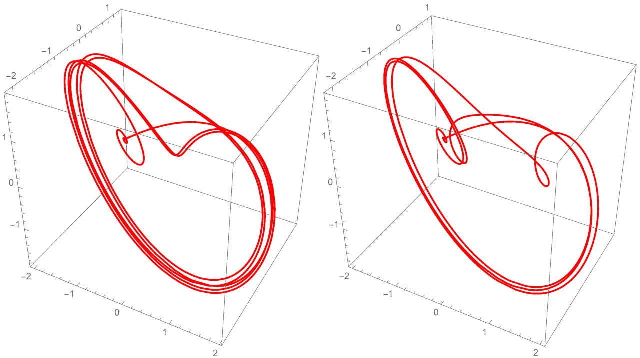
Theorem (L. Shilnikov, 1965)

x – saddle equilibrium

$\lambda = -\rho \pm i\omega$, $\rho > 0$ *and* γ *– eigenvalues of linearization of the vector field at x*

If $\rho/\gamma < 1$ *then near a homoclinic orbit there are countable many saddle periodic orbits near homoclinic orbit.*

Shilnikov homoclinic orbits

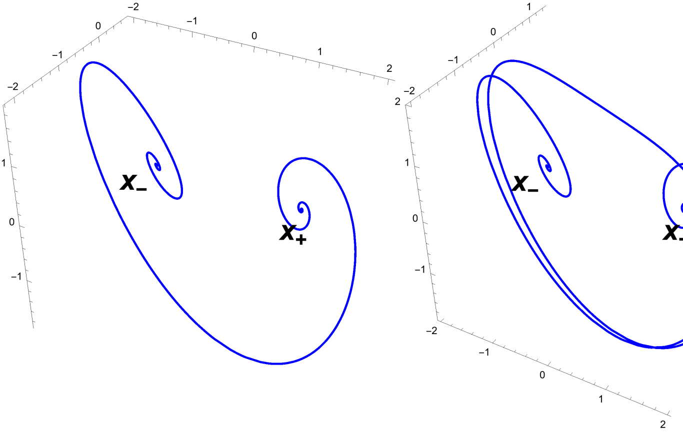


Theorem (L. Shilnikov, 1965)

x – saddle equilibrium
$\lambda = -\rho \pm i\omega$, $\rho > 0$ *and* γ *– eigenvalues of linearization of the vector field at x*

If $\rho/\gamma < 1$ *then near a homoclinic orbit there are countable many saddle periodic orbits near homoclinic orbit.*

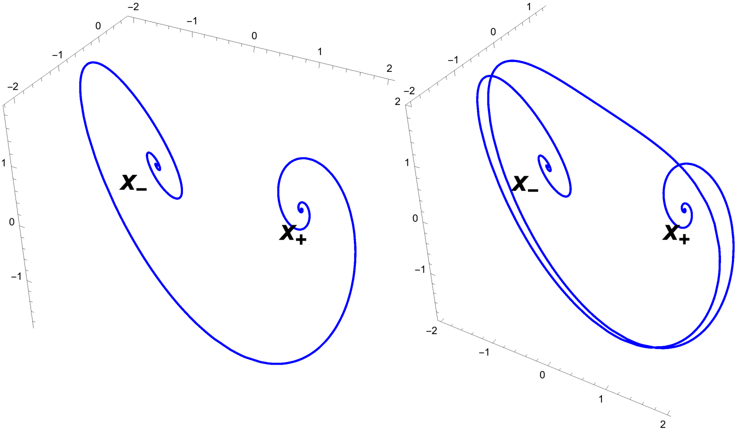
Transversal heteroclinic orbits



Geometry:

- x_- – 1D unstable, 2D stable
- x_+ – 1D stable, 2D unstable
- transversal intersection of two 2D manifolds (robust)
- possible infinity) of such connections for a single parameter

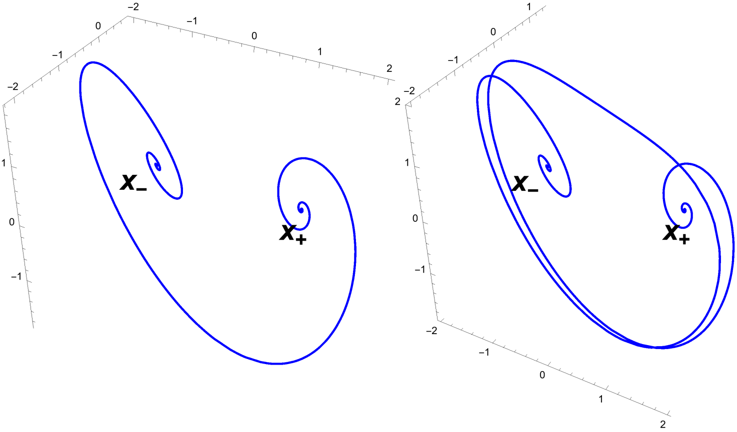
Transversal heteroclinic orbits



Geometry:

- x_- – 1D unstable, 2D stable
- x_+ – 1D stable, 2D unstable
- transversal intersection of two 2D manifolds (robust)
- possible infinity) of such connections for a single parameter

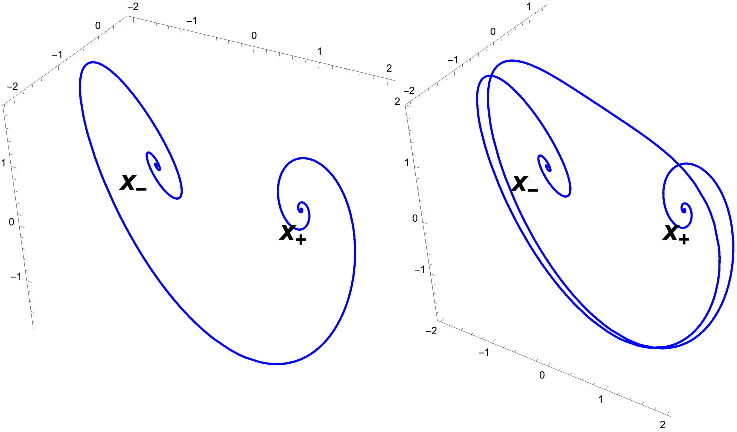
Transversal heteroclinic orbits



Geometry:

- x_- – 1D unstable, 2D stable
- x_+ – 1D stable, 2D unstable
- transversal intersection of two 2D manifolds (robust)
- possible infinity) of such connections for a single parameter

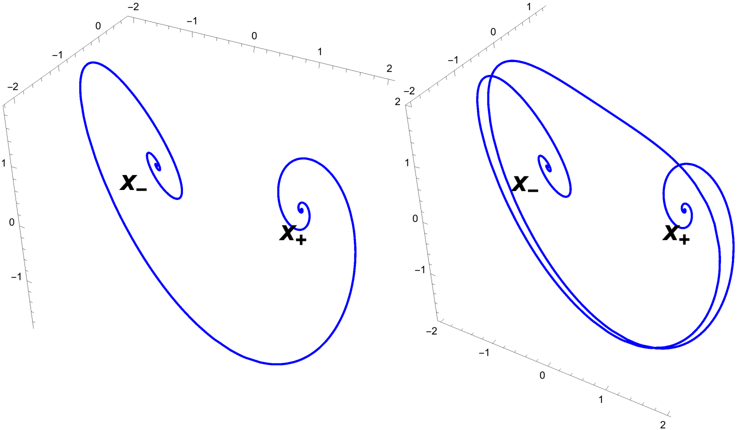
Transversal heteroclinic orbits



Geometry:

- x_- – 1D unstable, 2D stable
- x_+ – 1D stable, 2D unstable
- transversal intersection of two 2D manifolds (robust)
- possible infinity) of such connections for a single parameter

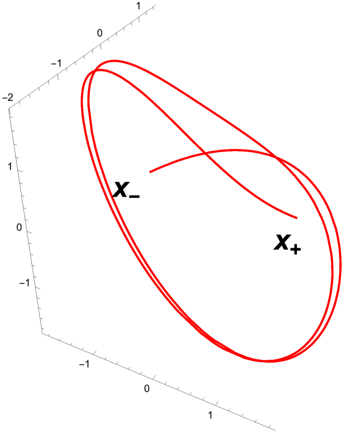
Transversal heteroclinic orbits

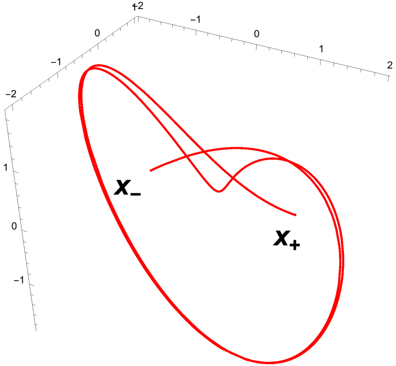


Geometry:

- x_- – 1D unstable, 2D stable
- x_+ – 1D stable, 2D unstable
- transversal intersection of two 2D manifolds (robust)
- possible infinity) of such connections for a single parameter

1D-1D heteroclinic orbit



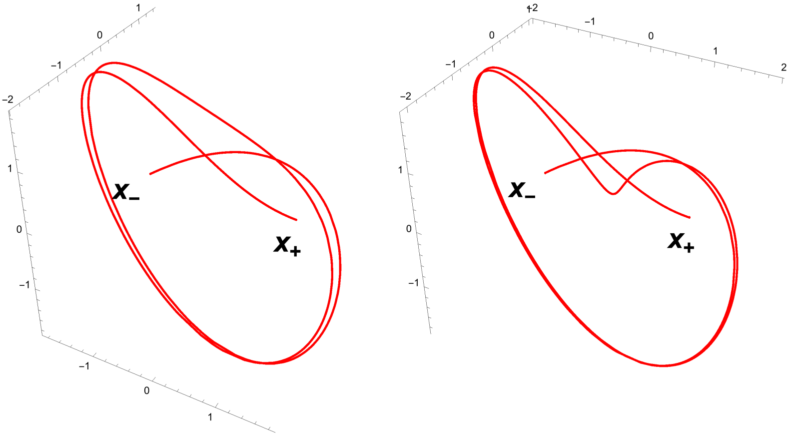


Geometry:

coincidence of branches of two 1D manifolds:

- x_- – 1D unstable
- x_+ – 1D stable
- codimension 2 bifurcation; in reversible systems codim 1

1D-1D heteroclinic orbit

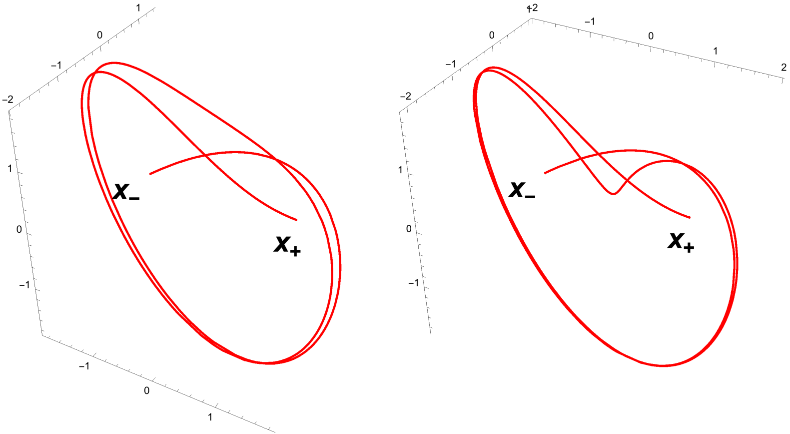


Geometry:

coincidence of branches of two 1D manifolds:

- x_- – 1D unstable
- x_+ – 1D stable
- codimension 2 bifurcation; in reversible systems codim 1

1D-1D heteroclinic orbit

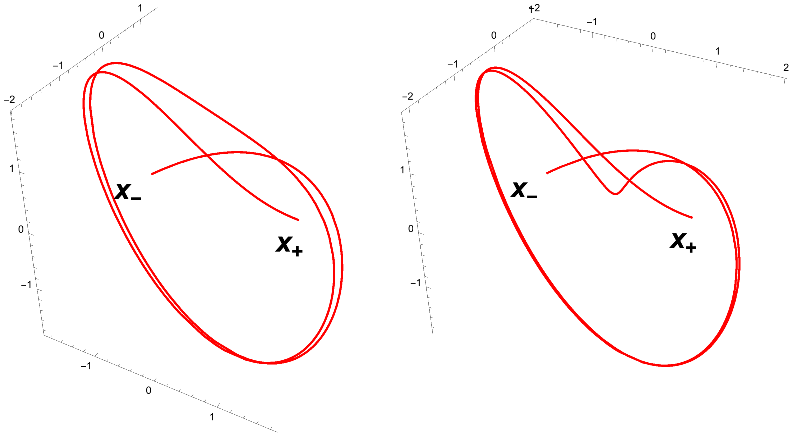


Geometry:

coincidence of branches of two 1D manifolds:

- x_- – 1D unstable
- x_+ – 1D stable
- codimension 2 bifurcation; in reversible systems codim 1

1D-1D heteroclinic orbit



Geometry:

coincidence of branches of two 1D manifolds:

- x_- – 1D unstable
- x_+ – 1D stable
- codimension 2 bifurcation; in reversible systems codim 1

Example (Michelson system)

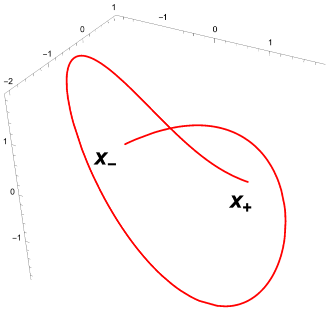
$$x' = y, \qquad y' = z, \qquad z' = c^2 - y - x^2/2$$

Theorem (Kuramoto and Tsuzuki, Prog. Theor. Phys.)

There is an explicit 1D-1D connection

$$x(t) = \alpha(-9\tanh(\beta t) + 11\tanh^3(\beta t)),$$

where $\alpha = 15\sqrt{11/19^3}$, $\beta = \frac{1}{2}\sqrt{11/19}$, $c = \alpha\sqrt{2} \approx 0.84952$.



Example (Michelson system)

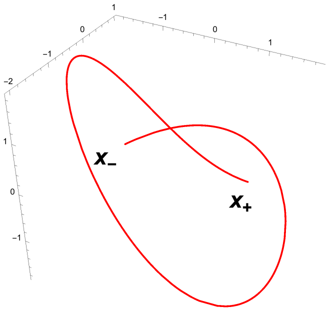
$$x' = y, \qquad y' = z, \qquad z' = c^2 - y - x^2/2$$

Theorem (Kuramoto and Tsuzuki, Prog. Theor. Phys.)

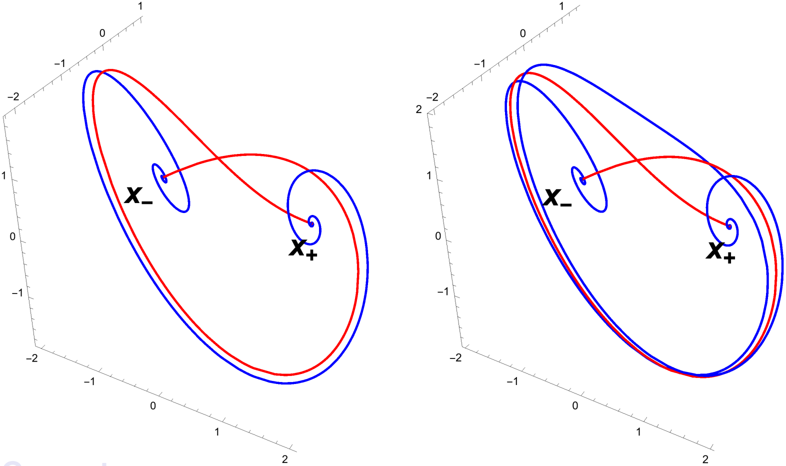
There is an explicit 1D-1D connection

$$x(t) = \alpha(-9\tanh(\beta t) + 11\tanh^3(\beta t)),$$

where $\alpha = 15\sqrt{11/19^3}$, $\beta = \frac{1}{2}\sqrt{11/19}$, $c = \alpha\sqrt{2} \approx 0.84952$.



Bykov cycle

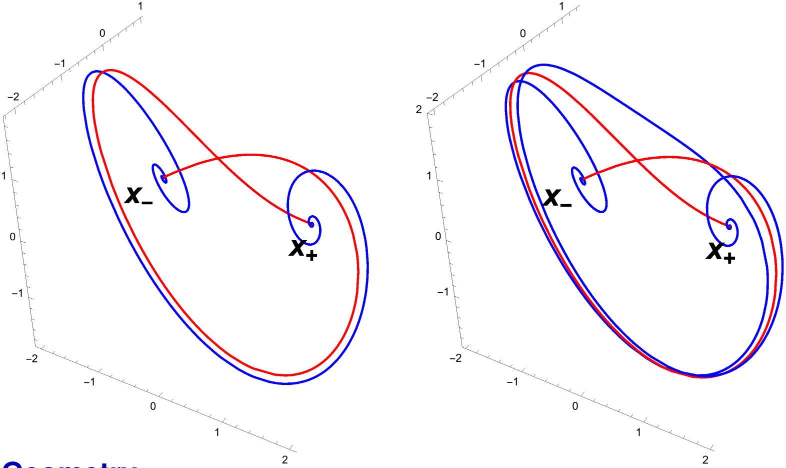


Geometry:

pair of heteroclinic orbits

- $x_- \to x_+$ – 1D-1D connection
- $x_+ \to x_-$ – transversal 2D-2D connection

Bykov cycle

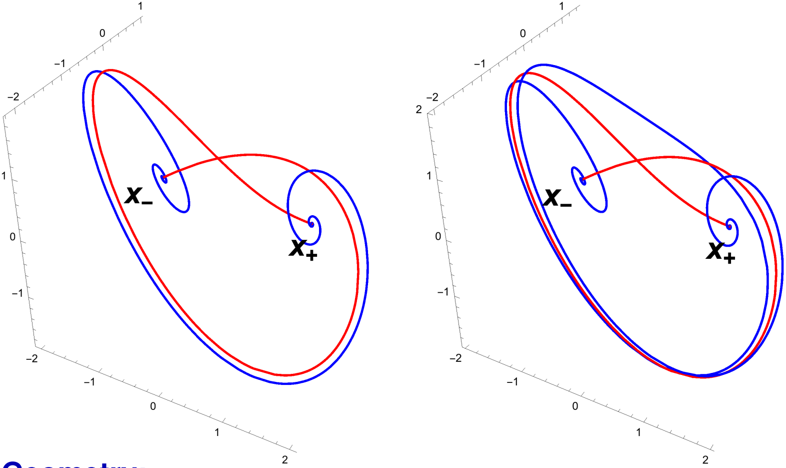


Geometry:

pair of heteroclinic orbits

- $x_- \rightarrow x_+$ – 1D-1D connection
- $x_+ \rightarrow x_-$ – transversal 2D-2D connection

Bykov cycle



Geometry:

pair of heteroclinic orbits

- $x_- \rightarrow x_+$ – 1D-1D connection
- $x_+ \rightarrow x_-$ – transversal 2D-2D connection

Some computer-assisted results for

Example (Michelson system)

$$x' = y, \qquad y' = z, \qquad z' = c^2 - y - x^2/2$$

Parameter range:

$$[c_{\min}, c_{\max}] = [0.8285, 0.861]$$

D. Wilczak, The Existence of Shilnikov Homoclinic Orbits in the Michelson System: A Computer Assisted Proof, Found. Comp. Math., Vol.6, No.4, 495–535 (2006).

Some computer-assisted results for

Example (Michelson system)

$$x' = y, \qquad y' = z, \qquad z' = c^2 - y - x^2/2$$

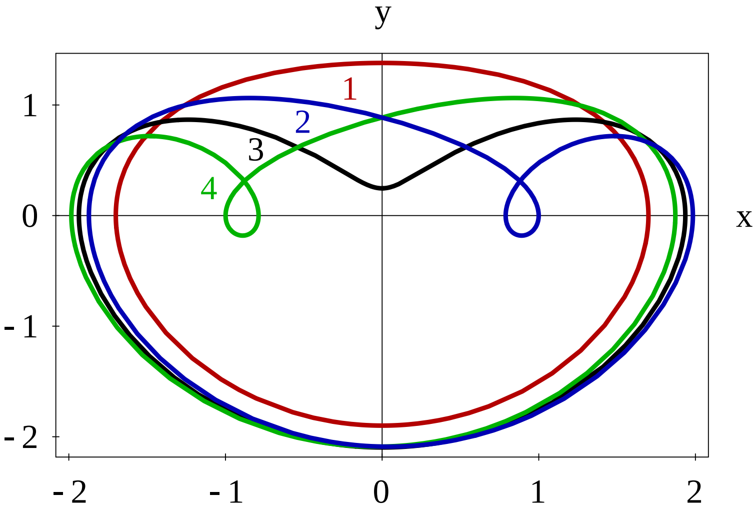
Parameter range:

$$[c_{\min}, c_{\max}] = [0.8285, 0.861]$$

D. Wilczak, The Existence of Shilnikov Homoclinic Orbits in the Michelson System: A Computer Assisted Proof, Found. Comp. Math., Vol.6, No.4, 495–535 (2006).

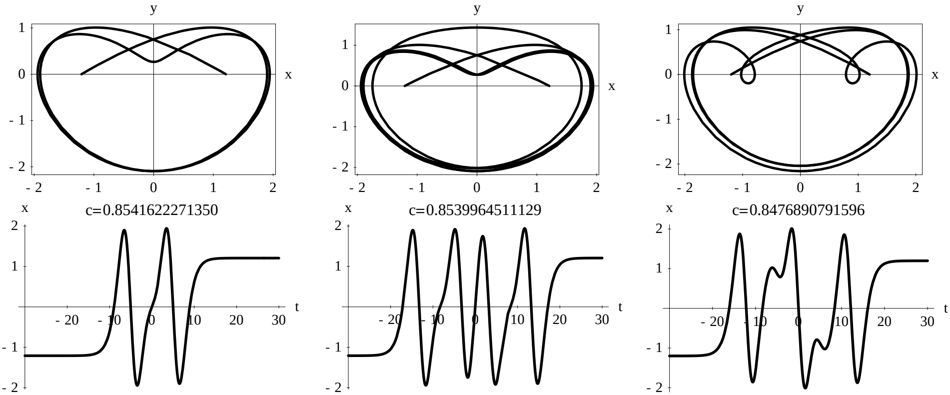
Theorem

For all parameter values $c \in [c_{\min}, c_{\max}]$ the Michelson system is Σ_4 chaotic.



Theorem

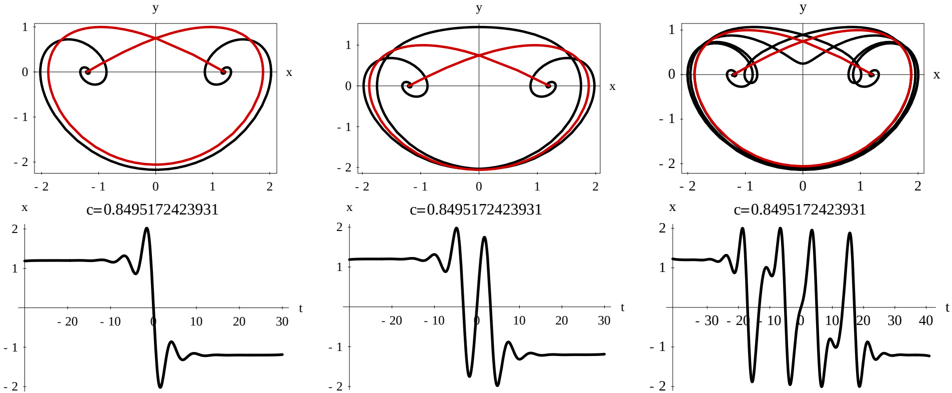
There exists a countable set of parameter values $C_h \subset [c_{\min}, c_{\max}]$, for which the Michelson system possesses 1D-1D heteroclinic solution.



Theorem

For all parameter values $c \in [c_{\min}, c_{\max}]$ there exist countable infinity of symmetric 2D-2D heteroclinic solutions.

In consequence, for countable parameter values C_h there is countable infinity of geometrically different Bykov cycles.

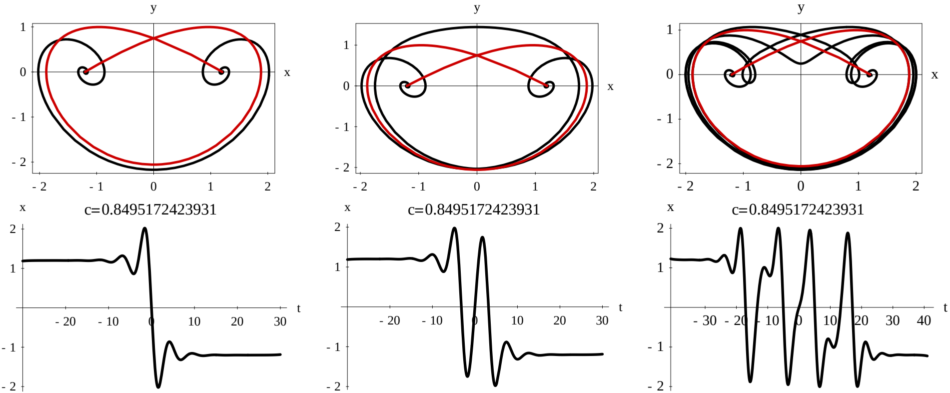


Three Bykov cycles for Kuramoto-Tsuzuki parameter $c = 15\sqrt{22/19^3}$

Theorem

For all parameter values $c \in [c_{\min}, c_{\max}]$ there exist countable infinity of symmetric 2D-2D heteroclinic solutions.

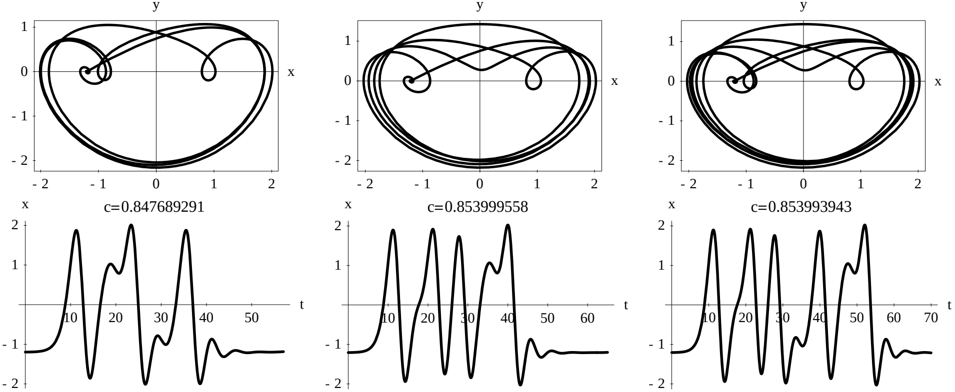
In consequence, for countable parameter values C_h there is countable infinity of geometrically different Bykov cycles.



Three Bykov cycles for Kuramoto-Tsuzuki parameter $c = 15\sqrt{22/19^3}$

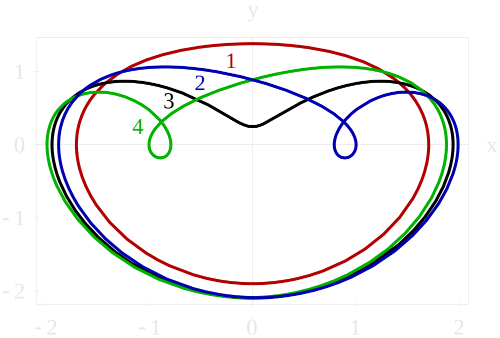
Theorem

There exists a countable set of parameter values $C_s \subset [c_{\min}, c_{\max}]$, for which the Michelson system possesses a pair of Shilnikov homoclinic solutions.



1D manifold

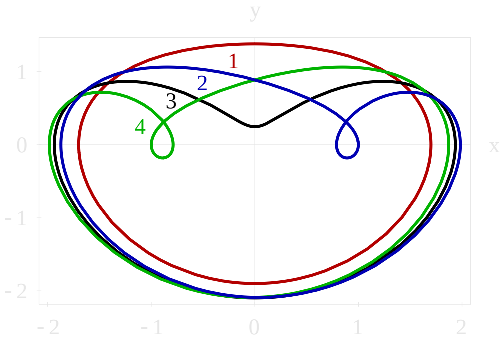
- leaves equilibrium
- follows four periodic orbits in arbitrary order for a finite (but arbitrarily large) time



- Shilnikov: converges back to equilibrium
- Bykov: intersects the symmetry line $y = 0$

1D manifold

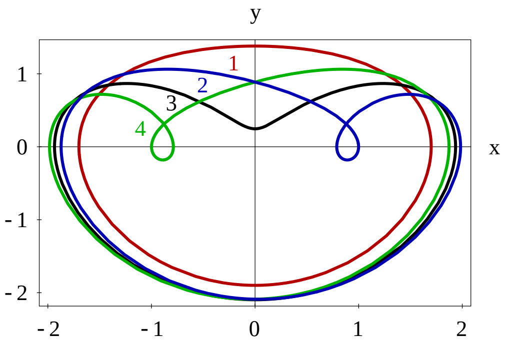
- leaves equilibrium
- follows four periodic orbits in arbitrary order for a finite (but arbitrarily large) time



- Shilnikov: converges back to equilibrium
- Bykov: intersects the symmetry line $y = 0$

1D manifold

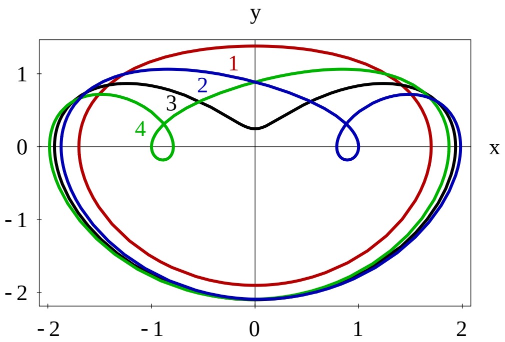
- leaves equilibrium
- follows four periodic orbits in arbitrary order for a finite (but arbitrarily large) time



- Shilnikov: converges back to equilibrium
- Bykov: intersects the symmetry line $y = 0$

1D manifold

- leaves equilibrium
- follows four periodic orbits in arbitrary order for a finite (but arbitrarily large) time



- Shilnikov: converges back to equilibrium
- Bykov: intersects the symmetry line $y = 0$

Computer-assisted proof of Σ_4 chaos

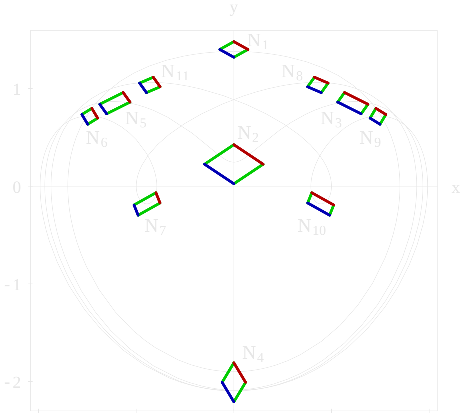
Methodology:

covering relations by Zgliczyński

(the same we used in the Rössler system)

Data:

Choose 11 parallelograms on $\Theta = \{(x, y, z) \in \mathbb{R}^3 : z = 0\}$.



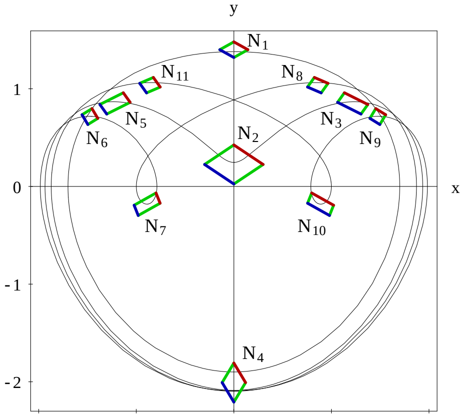
Methodology:

covering relations by Zgliczyński

(the same we used in the Rössler system)

Data:

Choose 11 parallelograms on $\Theta = \{(x, y, z) \in \mathbb{R}^3 : z = 0\}$.



Lemma (computer-assisted)

Put $C := [c_{\min}, c_{\max}]$.
The Poincaré map $P : C \times \Theta \to \Theta$ *is well defined and continuous on* $C \times \bigcup_{i=1}^{11} N_i$. *Moreover, for all* $c \in C$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4},$$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_5} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_2} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_3} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4},$$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_6} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_7} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_8} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4},$$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_{11}} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_{10}} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_9} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4}.$$

Lemma (computer-assisted)

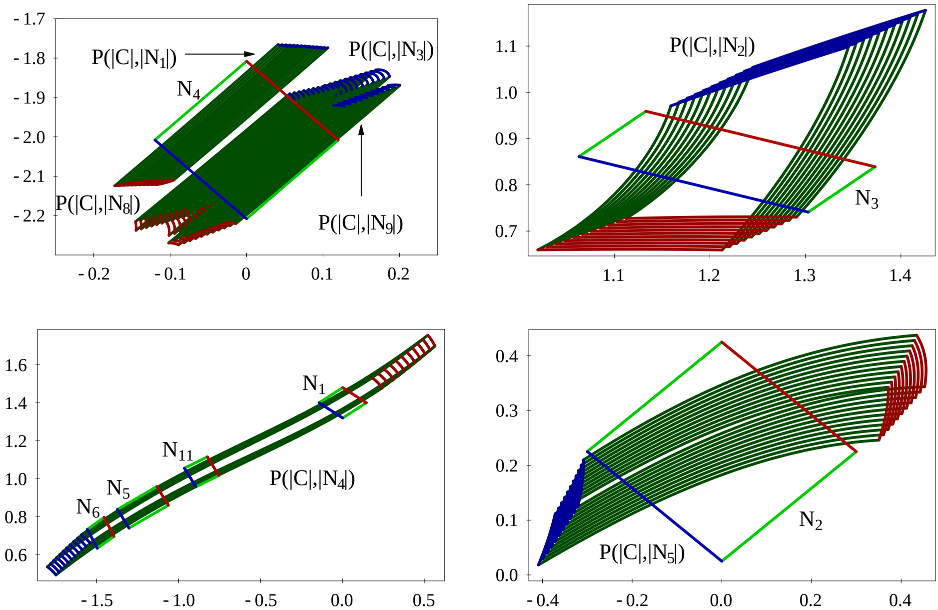
Put $C := [c_{\min}, c_{\max}]$.
The Poincaré map $P : C \times \Theta \to \Theta$ is well defined and continuous on $C \times \bigcup_{i=1}^{11} N_i$. Moreover, for all $c \in C$

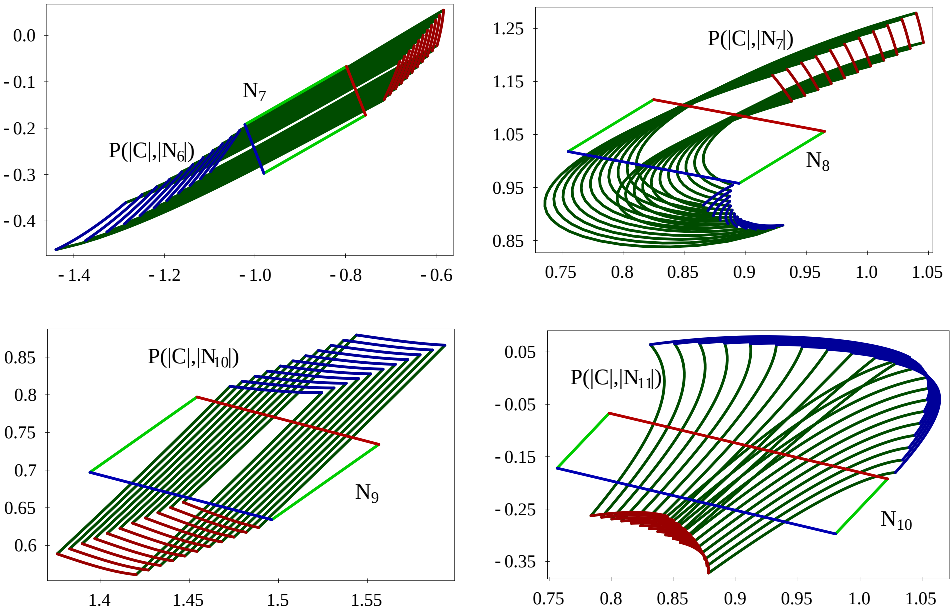
$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4},$$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_5} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_2} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_3} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4},$$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_6} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_7} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_8} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4},$$

$$\mathbf{N_4} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_{11}} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_{10}} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_9} \overset{\mathbf{P(c,\cdot)}}{\Longrightarrow} \mathbf{N_4}.$$


P(|C|,|N1|)
P(|C|,|N3|)
N4
P(|C|,|N8|)
P(|C|,|N9|)
P(|C|,|N2|)
N3
N1
N11
N5
N6
P(|C|,|N4|)
N2
P(|C|,|N5|)



P(|C|,|N6|)
N7
P(|C|,|N7|)
N8
P(|C|,|N10|)
N9
P(|C|,|N11|)
N10

New theory

Computer-assisted proofs for

- Shilnikov homoclinic orbits
- heteroclinic Bykov cycles

require new **“good” abstract theorems**.

Description of the method

Definition

***h*-set** N is an object consisting of

- $|N|$ - compact subset of $\mathbb{R}^n$ (called **support**)
- $u(N), s(N) \in \{0, 1, 2, \dots\}$,
  such that $u(N) + s(N) = n$
- a homeomorphism $c_N : \mathbb{R}^n \to \mathbb{R}^n = \mathbb{R}^{u(N)} \times \mathbb{R}^{s(N)}$ such that

$$c_N(|N|) = \overline{B_{u(N)}}(0, 1) \times \overline{B_{s(N)}}(0, 1).$$

Description of the method

Definition

***h*-set** N is an object consisting of

- $|N|$ - compact subset of $\mathbb{R}^n$ (called **support**)
- $u(N), s(N) \in \{0, 1, 2, \ldots\}$,
  such that $u(N) + s(N) = n$
- a homeomorphism $c_N : \mathbb{R}^n \to \mathbb{R}^n = \mathbb{R}^{u(N)} \times \mathbb{R}^{s(N)}$ such that

$$c_N(|N|) = \overline{B_{u(N)}}(0, 1) \times \overline{B_{s(N)}}(0, 1).$$

Description of the method

Definition

***h*-set** N is an object consisting of

- $|N|$ - compact subset of $\mathbb{R}^n$ (called **support**)
- $u(N), s(N) \in \{0, 1, 2, \dots\}$,
  such that $u(N) + s(N) = n$
- a homeomorphism $c_N : \mathbb{R}^n \to \mathbb{R}^n = \mathbb{R}^{u(N)} \times \mathbb{R}^{s(N)}$ such that

$$c_N(|N|) = \overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1).$$

Definition

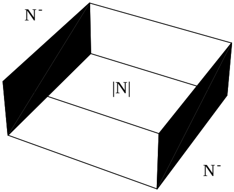
***h*-set** N is an object consisting of

- $|N|$ - compact subset of $\mathbb{R}^n$ (called **support**)
- $u(N), s(N) \in \{0, 1, 2, \ldots\}$,
  such that $u(N) + s(N) = n$
- a homeomorphism $c_N : \mathbb{R}^n \to \mathbb{R}^n = \mathbb{R}^{u(N)} \times \mathbb{R}^{s(N)}$ such that

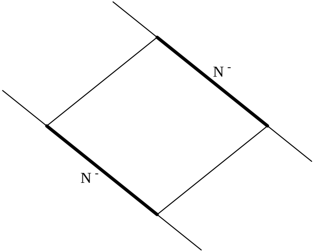
$$c_N(|N|) = \overline{B_{u(N)}}(0, 1) \times \overline{B_{s(N)}}(0, 1).$$

Description of the method

$u(N) = 1$ **and** $s(N) = 2$



$u(N) = s(N) = 1$



Notation:

$$N_c = \overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1),$$

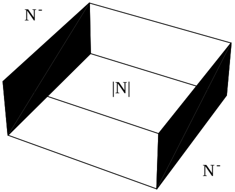
$$N_c^- = \partial \overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1)$$

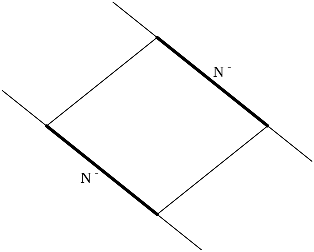
$$N_c^+ = \overline{B_{u(N)}}(0,1) \times \partial \overline{B_{s(N)}}(0,1)$$

$$N^- = c_N^{-1}(N_c^-), \quad N^+ = c_N^{-1}(N_c^+)$$

Description of the method

$u(N) = 1$ **and** $s(N) = 2$ $\qquad$ $u(N) = s(N) = 1$





Notation:

$$N_c = \overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1),$$

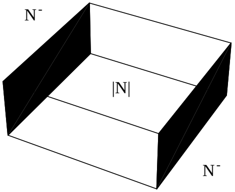
$$N_c^- = \partial\overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1)$$

$$N_c^+ = \overline{B_{u(N)}}(0,1) \times \partial\overline{B_{s(N)}}(0,1)$$

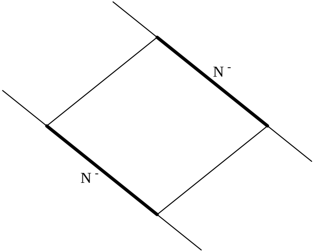
$$N^- = c_N^{-1}(N_c^-), \quad N^+ = c_N^{-1}(N_c^+)$$

Description of the method

$u(N) = 1$ **and** $s(N) = 2$



$u(N) = s(N) = 1$



Notation:

$$N_c = \overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1),$$

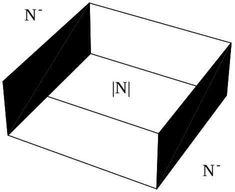
$$N_c^- = \partial\overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1)$$

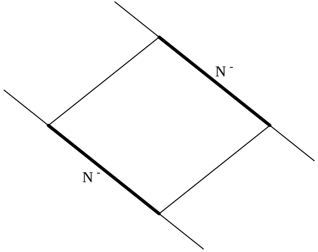
$$N_c^+ = \overline{B_{u(N)}}(0,1) \times \partial\overline{B_{s(N)}}(0,1)$$

$$N^- = c_N^{-1}(N_c^-), \quad N^+ = c_N^{-1}(N_c^+)$$

Description of the method

$u(N) = 1$ **and** $s(N) = 2$ $\qquad$ $u(N) = s(N) = 1$





Notation:

$$N_c = \overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1),$$

$$N_c^- = \partial\overline{B_{u(N)}}(0,1) \times \overline{B_{s(N)}}(0,1)$$

$$N_c^+ = \overline{B_{u(N)}}(0,1) \times \partial\overline{B_{s(N)}}(0,1)$$

$$N^- = c_N^{-1}(N_c^-), \quad N^+ = c_N^{-1}(N_c^+)$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

N **f-covers** M ($N \overset{f}{\Longrightarrow} M$) iff

1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that

$$\begin{aligned} h(0,\cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$

2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that

$$\begin{aligned} h_1(p,q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

N **f-covers** M ($N \overset{f}{\Longrightarrow} M$) iff

1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that

$$\begin{aligned} h(0,\cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$

2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that

$$\begin{aligned} h_1(p,q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

N f-covers M ($N \overset{f}{\Longrightarrow} M$) iff

1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that
$$\begin{aligned} h(0, \cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$
2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that
$$\begin{aligned} h_1(p, q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

N f-covers M ($N \overset{f}{\Longrightarrow} M$) iff

1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that

$$\begin{aligned} h(0,\cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$

2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that

$$\begin{aligned} h_1(p,q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

N f-covers M ($N \overset{f}{\Longrightarrow} M$) iff

1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that

$$\begin{aligned} h(0,\cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$

2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that

$$\begin{aligned} h_1(p,q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

N f-covers M ($N \overset{f}{\Longrightarrow} M$) iff

1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that

$$\begin{aligned} h(0,\cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$

2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that

$$\begin{aligned} h_1(p,q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

Settings:

N, M h-sets with $u(N) = u(M) = u$

$f : |N| \to \mathbb{R}^n$ – continuous,

$f_c = c_M \circ f \circ c_N^{-1} : N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$.

Definition (Gidea, Zgliczyński 2003, DW 2006)

***N f*-covers *M* ($N \overset{f}{\Longrightarrow} M$)** iff

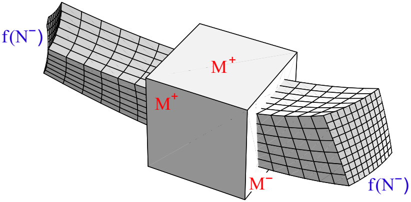
1. There exists $h : [0,1] \times N_c \to \mathbb{R}^u \times \mathbb{R}^{s(M)}$ such that

$$\begin{aligned} h(0,\cdot) &= f_c, \\ h([0,1], N_c^-) \cap M_c &= \emptyset, \\ h([0,1], N_c) \cap M_c^+ &= \emptyset. \end{aligned}$$

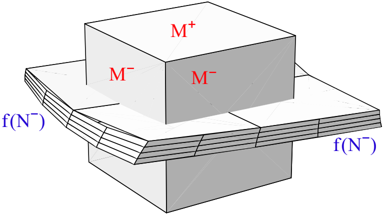
2. There exists a linear map $A : \mathbb{R}^u \to \mathbb{R}^u$ such that

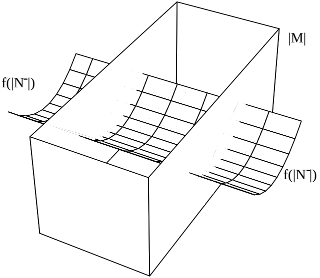
$$\begin{aligned} h_1(p,q) &= (A(p), 0), \\ A(\partial B_u(0,1)) &\subset \mathbb{R}^u \setminus \overline{B_u}(0,1) \end{aligned}$$

$u = 1, s = 2$



$u = 2, s = 1$





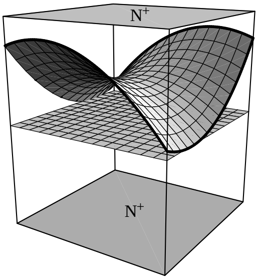
$u = 1, s(N) = 1, s(M) = 2$

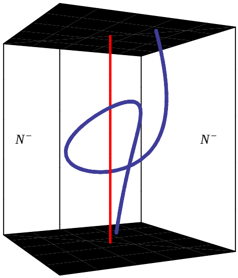
$b : \overline{B_{u(N)}} \to |N|$ – continuous.
Put $b_c = c_N \circ b$.

Definition

b – **a horizontal disc in N** if there exists a homotopy $h : [0,1] \times \overline{B_{u(N)}} \to N_c$ such that

$$
\begin{aligned}
h_0 &= b_c \\
h_1(x) &= (x, 0), \quad \text{for } x \in \overline{B_{u(N)}} \\
h(t,x) &\in N_c^-, \quad \text{for } t \in [0,1] \text{ and } x \in \partial \overline{B_{u(N)}}
\end{aligned}
$$





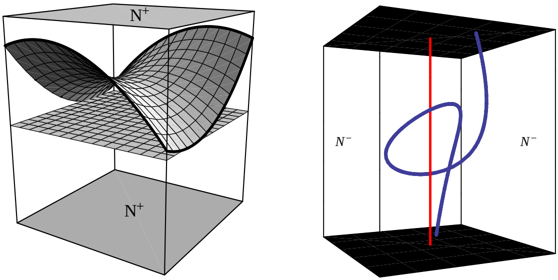
horizontal and vertical discs in an h-set with $u = 2$, $s = 1$

$b : \overline{B_{u(N)}} \to |N|$ – continuous.
Put $b_c = c_N \circ b$.

Definition

b – **a horizontal disc in N** if there exists a homotopy $h : [0,1] \times \overline{B_{u(N)}} \to N_c$ such that

$$\begin{aligned} h_0 &= b_c \\ h_1(x) &= (x, 0), \quad \text{for } x \in \overline{B_{u(N)}} \\ h(t,x) &\in N_c^-, \quad \text{for } t \in [0,1] \text{ and } x \in \partial \overline{B_{u(N)}} \end{aligned}$$



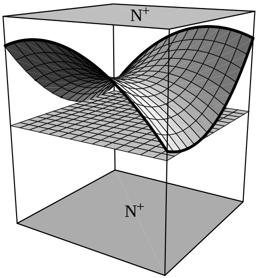
horizontal and vertical discs in an h-set with $u = 2$, $s = 1$

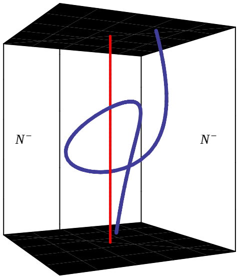
$b : \overline{B_{u(N)}} \to |N|$ – continuous.
Put $b_c = c_N \circ b$.

Definition

b – **a horizontal disc in N** if there exists a homotopy $h : [0,1] \times \overline{B_{u(N)}} \to N_c$ such that

$$
\begin{aligned}
h_0 &= b_c \\
h_1(x) &= (x, 0), \quad \text{for } x \in \overline{B_{u(N)}} \\
h(t,x) &\in N_c^-, \quad \text{for } t \in [0,1] \text{ and } x \in \partial\overline{B_{u(N)}}
\end{aligned}
$$





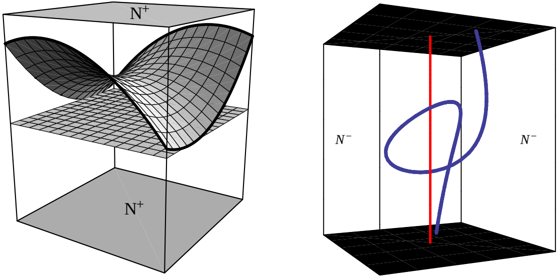
horizontal and vertical discs in an h-set with $u = 2$, $s = 1$

$b : \overline{B_{u(N)}} \to |N|$ – continuous.
Put $b_c = c_N \circ b$.

Definition

b – **a horizontal disc in N** if there exists a homotopy
$h : [0,1] \times \overline{B_{u(N)}} \to N_c$ such that

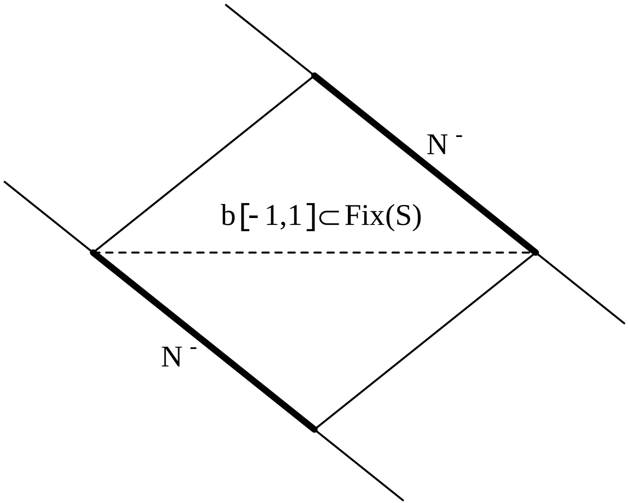
$$\begin{aligned} h_0 &= b_c \\ h_1(x) &= (x, 0), \quad \text{for } x \in \overline{B_{u(N)}} \\ h(t,x) &\in N_c^-, \quad \text{for } t \in [0,1] \text{ and } x \in \partial \overline{B_{u(N)}} \end{aligned}$$



horizontal and vertical discs in an h-set with $u = 2$, $s = 1$

Important in reversible systems:

S – symmetry



b is both horizontal and vertical disc in N

Theorem (Main topological result)

$C, N_0, N_1, \ldots, N_k$, $k > 0$ h-sets, such that
$\dim(C) = u(C) = u(N_0) = \cdots = u(N_k)$.
$W : C \to \mathbb{R}^{\dim(N_0)}$ *– continuous*
$f_i : C \times N_i \to \mathbb{R}^{\dim(N_{i+1})}$, $i = 0, \ldots, k-1$ *continuous,*
$v : B_{s(N_k)} \to N_k$ *vertical disc in* N_k.

If $\mathbf{C} \overset{\mathbf{W}}{\Longrightarrow} \mathbf{N_0}$ *and for all* $c \in C$

$$\mathbf{N_0} \overset{\mathbf{f_0(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{f_1(c,\cdot)}}{\Longrightarrow} \cdots \overset{\mathbf{f_{k-1}(c,\cdot)}}{\Longrightarrow} \mathbf{N_k}$$

then there exists $c_0 \in C$ *such that*

$$\begin{aligned} W(c_0) &\in N_0, \\ (f_i(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in N_{i+1}, \quad i = 0, \ldots, k-2 \\ (f_{k-1}(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in v(B_{s(N_k)}). \end{aligned}$$

Theorem (Main topological result)

$C, N_0, N_1, \ldots, N_k$, $k > 0$ h-sets, such that
$\dim(C) = u(C) = u(N_0) = \cdots = u(N_k)$.
$W : C \to \mathbb{R}^{\dim(N_0)}$ *– continuous*
$f_i : C \times N_i \to \mathbb{R}^{\dim(N_{i+1})}$, $i = 0, \ldots, k-1$ *continuous,*
$v : B_{s(N_k)} \to N_k$ *vertical disc in* N_k.

If $\mathbf{C} \overset{\mathbf{W}}{\Longrightarrow} \mathbf{N_0}$ *and for all* $c \in C$

$$\mathbf{N_0} \overset{\mathbf{f_0(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{f_1(c,\cdot)}}{\Longrightarrow} \ldots \overset{\mathbf{f_{k-1}(c,\cdot)}}{\Longrightarrow} \mathbf{N_k}$$

then there exists $c_0 \in C$ *such that*

$$\begin{aligned} W(c_0) &\in N_0, \\ (f_i(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in N_{i+1}, \quad i = 0, \ldots, k-2 \\ (f_{k-1}(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in v(B_{s(N_k)}). \end{aligned}$$

Theorem (Main topological result)

$C, N_0, N_1, \ldots, N_k$, $k > 0$ h-sets, such that
$\dim(C) = u(C) = u(N_0) = \cdots = u(N_k)$.
$W : C \to \mathbb{R}^{\dim(N_0)}$ *– continuous*
$f_i : C \times N_i \to \mathbb{R}^{\dim(N_{i+1})}$, $i = 0, \ldots, k-1$ *continuous,*
$v : B_{s(N_k)} \to N_k$ *vertical disc in* N_k.

If $\mathbf{C} \overset{\mathbf{W}}{\Longrightarrow} \mathbf{N_0}$ *and for all* $c \in C$

$$\mathbf{N_0} \overset{\mathbf{f_0(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{f_1(c,\cdot)}}{\Longrightarrow} \ldots \overset{\mathbf{f_{k-1}(c,\cdot)}}{\Longrightarrow} \mathbf{N_k}$$

then there exists $c_0 \in C$ *such that*

$$\begin{aligned} W(c_0) &\in N_0, \\ (f_i(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in N_{i+1}, \quad i = 0, \ldots, k-2 \\ (f_{k-1}(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in v(B_{s(N_k)}). \end{aligned}$$

Theorem (Main topological result)

$C, N_0, N_1, \dots, N_k$, $k > 0$ h-sets, such that
$\dim(C) = u(C) = u(N_0) = \cdots = u(N_k)$.
$W : C \to \mathbb{R}^{\dim(N_0)}$ *– continuous*
$f_i : C \times N_i \to \mathbb{R}^{\dim(N_{i+1})}$, $i = 0, \dots, k-1$ *continuous,*
$v : B_{s(N_k)} \to N_k$ *vertical disc in* N_k.

If $\mathbf{C} \overset{\mathbf{W}}{\Longrightarrow} \mathbf{N_0}$ *and for all* $c \in C$

$$\mathbf{N_0} \overset{\mathbf{f_0(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{f_1(c,\cdot)}}{\Longrightarrow} \dots \overset{\mathbf{f_{k-1}(c,\cdot)}}{\Longrightarrow} \mathbf{N_k}$$

then there exists $c_0 \in C$ *such that*

$$\begin{aligned} W(c_0) &\in N_0, \\ (f_i(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in N_{i+1}, \quad i = 0, \dots, k-2 \\ (f_{k-1}(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in v(B_{s(N_k)}). \end{aligned}$$

Theorem (Main topological result)

$C, N_0, N_1, \dots, N_k$, $k > 0$ h-sets, such that
$\dim(C) = u(C) = u(N_0) = \cdots = u(N_k)$.
$W : C \to \mathbb{R}^{\dim(N_0)}$ *– continuous*
$f_i : C \times N_i \to \mathbb{R}^{\dim(N_{i+1})}$, $i = 0, \dots, k-1$ *continuous,*
$v : B_{s(N_k)} \to N_k$ *vertical disc in* N_k.

If $\mathbf{C} \overset{\mathbf{W}}{\Longrightarrow} \mathbf{N_0}$ *and for all* $c \in C$

$$\mathbf{N_0} \overset{\mathbf{f_0(c,\cdot)}}{\Longrightarrow} \mathbf{N_1} \overset{\mathbf{f_1(c,\cdot)}}{\Longrightarrow} \dots \overset{\mathbf{f_{k-1}(c,\cdot)}}{\Longrightarrow} \mathbf{N_k}$$

then there exists $c_0 \in C$ *such that*

$$\begin{aligned} W(c_0) &\in N_0, \\ (f_i(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in N_{i+1}, \quad i = 0, \dots, k-2 \\ (f_{k-1}(c_0,\cdot) \circ \cdots \circ f_0(c_0,\cdot))(W(c_0)) &\in v(B_{s(N_k)}). \end{aligned}$$

Interpretation:

C – parameters of the system
$W(c)$ – property of the system
(as a point in the phase space)

For some parameter c_0:

trajectory of $W(c_0)$ intersects

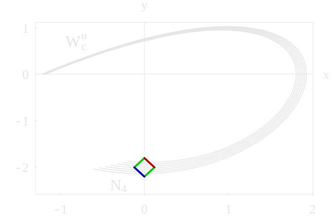
$$N_1, N_2, \ldots, N_k$$

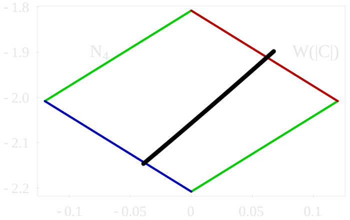
In the last set:

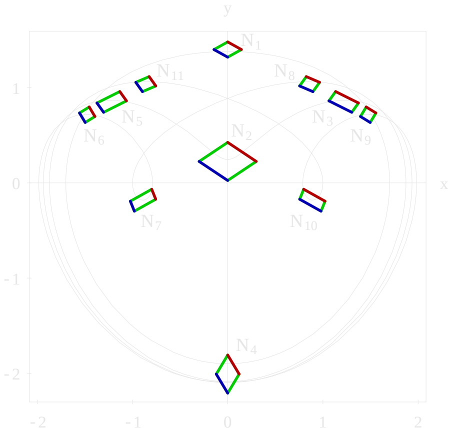
trajectory of $W(c_0)$ hits vertical disc.

1D-1D heteroclinic connections

- $W(c)$ – second cut of the manifold with section
- propagate it along four periodic orbits
- slice of symmetry line $y = 0$ is vertical disc in N_1, N_2, N_4!





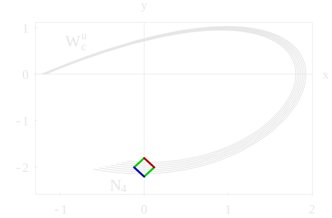


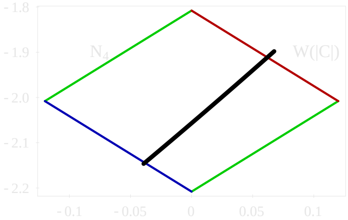
Idea:

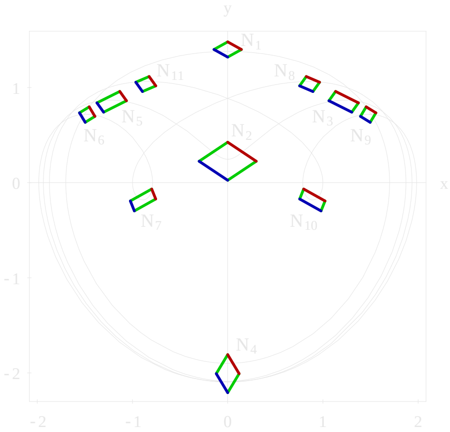
$$\text{manifold} \xrightarrow{W} N_4 \xrightarrow{P} \cdots \text{follow 4PO} \cdots \xrightarrow{P} \{N_1, N_2, N_4\} \cap \{y = 0\}$$

1D-1D heteroclinic connections

- $W(c)$ – second cut of the manifold with section
- propagate it along four periodic orbits
- slice of symmetry line $y = 0$ is vertical disc in N_1, N_2, N_4!





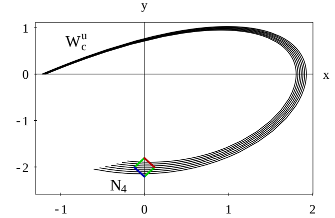


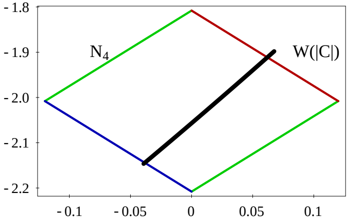
Idea:

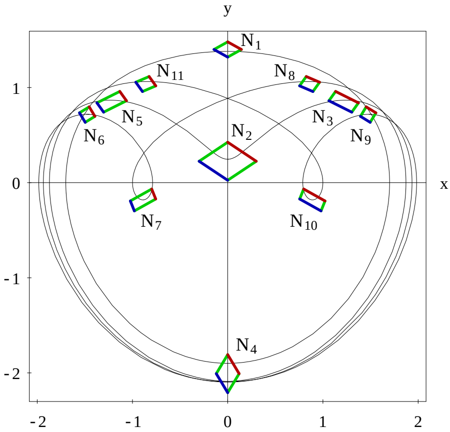
$$\text{manifold} \overset{W}{\Longrightarrow} N_4 \overset{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \overset{P}{\Longrightarrow} \{N_1, N_2, N_4\} \cap \{y = 0\}$$

1D-1D heteroclinic connections

- $W(c)$ – second cut of the manifold with section
- propagate it along four periodic orbits
- slice of symmetry line $y = 0$ is vertical disc in N_1, N_2, N_4!





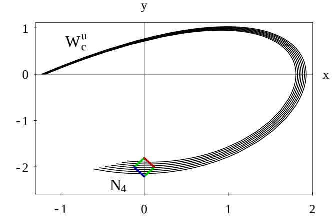


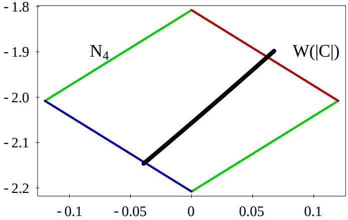
Idea:

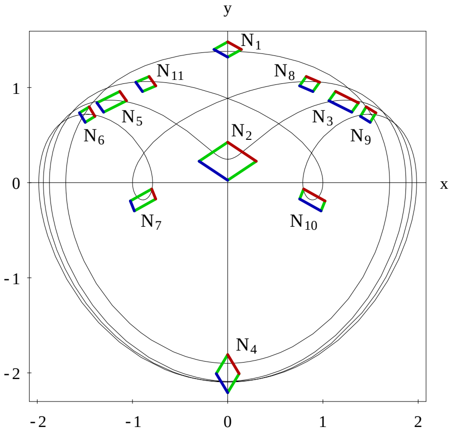
manifold $\xrightarrow{W} N_4 \xrightarrow{P} \cdots$ follow 4PO $\cdots \xrightarrow{P} \{N_1, N_2, N_4\} \cap \{y = 0\}$

1D-1D heteroclinic connections

- $W(c)$ – second cut of the manifold with section
- propagate it along four periodic orbits
- slice of symmetry line $y = 0$ is vertical disc in N_1, N_2, N_4!



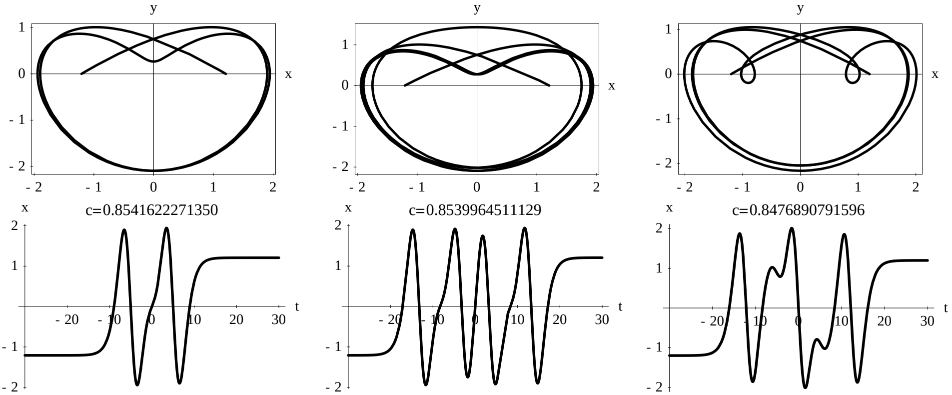




Idea:

$$\text{manifold} \overset{W}{\Longrightarrow} N_4 \overset{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \overset{P}{\Longrightarrow} \{N_1, N_2, N_4\} \cap \{y = 0\}$$

Some heteroclinic connections



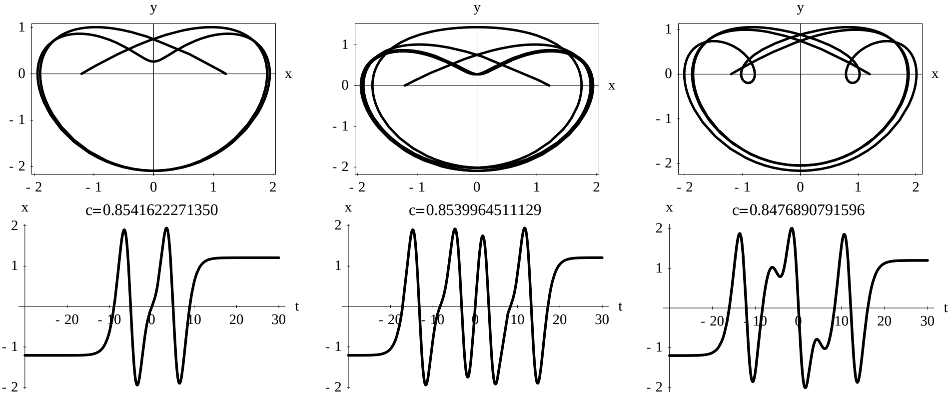
Computational aspects:

- estimate invariant manifold near equilibrium

  (there is an explicit Lyapunov function in the half-plane)

- integrate manifold until Poincaré section and compute intersection
- validate covering relations between sets $N_1, \ldots, N_{11}$

Some heteroclinic connections

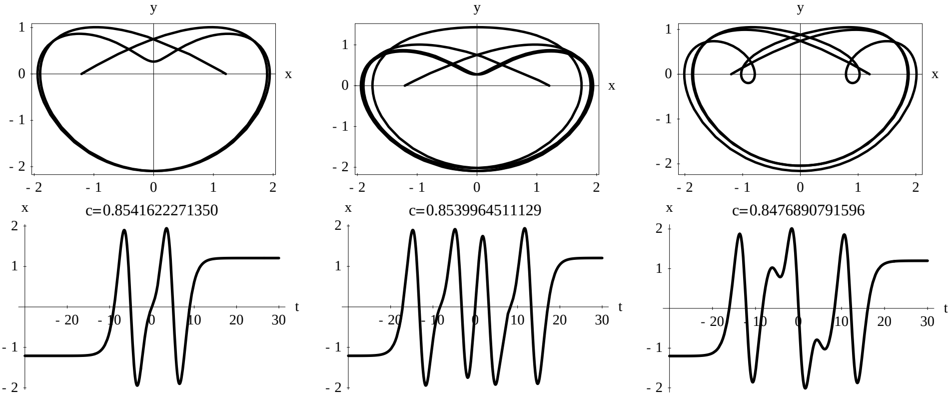


Computational aspects:

- estimate invariant manifold near equilibrium

  (there is an explicit Lyapunov function in the half-plane)
- integrate manifold until Poincaré section and compute intersection
- validate covering relations between sets $N_1, \ldots, N_{11}$

Some heteroclinic connections



Computational aspects:

- estimate invariant manifold near equilibrium

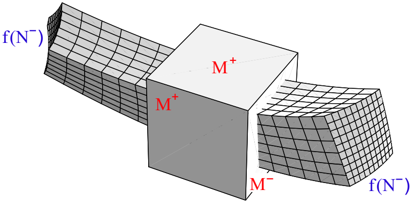
  (there is an explicit Lyapunov function in the half-plane)
- integrate manifold until Poincaré section and compute intersection
- validate covering relations between sets $N_1, \ldots, N_{11}$

2D-2D transversal connections

Idea:

$$\{y=0\}\cap\{N_1,N_2,N_4\} \stackrel{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \stackrel{P}{\Longrightarrow} N_4 \stackrel{P}{\Longrightarrow} H \stackrel{P}{\Longrightarrow} H$$

- slice of $y=0$ is horizontal disc in N_1, N_2, N_4
- transport it following four periodic orbit for some finite time and stop in N_4
- construct H – 3D set centered at equilibrium and spanned on eigenvectors
- check $H \stackrel{\varphi(T,\cdot)}{\Longrightarrow} H$, where φ flow for some T



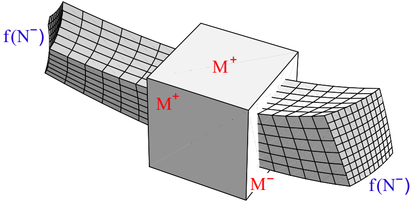
- check covering $N_4 \stackrel{\varphi(T,\cdot)}{\Longrightarrow} H$ for some T

2D-2D transversal connections

Idea:

$$\{y=0\}\cap\{N_1,N_2,N_4\} \overset{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \overset{P}{\Longrightarrow} N_4 \overset{P}{\Longrightarrow} H \overset{P}{\Longrightarrow} H$$

- slice of $y=0$ is horizontal disc in N_1, N_2, N_4
- transport it following four periodic orbit for some finite time and stop in N_4
- construct H – 3D set centered at equilibrium and spanned on eigenvectors
- check $H \overset{\varphi(T,\cdot)}{\Longrightarrow} H$, where φ flow for some T



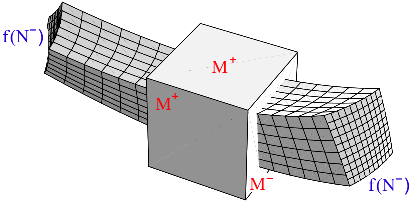
- check covering $N_4 \overset{\varphi(T,\cdot)}{\Longrightarrow} H$ for some T

2D-2D transversal connections

Idea:

$$\{y=0\}\cap\{N_1, N_2, N_4\} \stackrel{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \stackrel{P}{\Longrightarrow} N_4 \stackrel{P}{\Longrightarrow} H \stackrel{P}{\Longrightarrow} H$$

- slice of $y=0$ is horizontal disc in N_1, N_2, N_4
- transport it following four periodic orbit for some finite time and stop in N_4
- construct H – 3D set centered at equilibrium and spanned on eigenvectors
- check $H \stackrel{\varphi(T,\cdot)}{\Longrightarrow} H$, where φ flow for some T



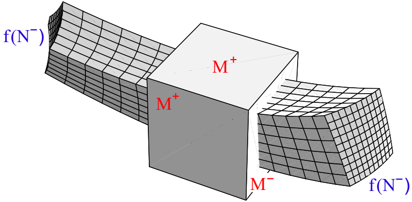
- check covering $N_4 \stackrel{\varphi(T,\cdot)}{\Longrightarrow} H$ for some T

2D-2D transversal connections

Idea:

$$\{y=0\}\cap\{N_1, N_2, N_4\} \overset{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \overset{P}{\Longrightarrow} N_4 \overset{P}{\Longrightarrow} H \overset{P}{\Longrightarrow} H$$

- slice of $y=0$ is horizontal disc in N_1, N_2, N_4
- transport it following four periodic orbit for some finite time and stop in N_4
- construct H – 3D set centered at equilibrium and spanned on eigenvectors
- check $H \overset{\varphi(T,\cdot)}{\Longrightarrow} H$, where φ flow for some T



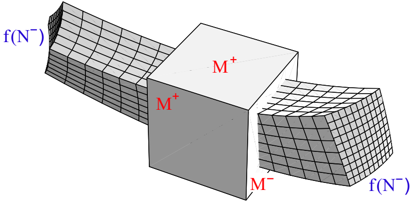
- check covering $N_4 \overset{\varphi(T,\cdot)}{\Longrightarrow} H$ for some T

2D-2D transversal connections

Idea:

$$\{y=0\}\cap\{N_1, N_2, N_4\} \overset{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \overset{P}{\Longrightarrow} N_4 \overset{P}{\Longrightarrow} H \overset{P}{\Longrightarrow} H$$

- slice of $y=0$ is horizontal disc in N_1, N_2, N_4
- transport it following four periodic orbit for some finite time and stop in N_4
- construct H – 3D set centered at equilibrium and spanned on eigenvectors
- check $H \overset{\varphi(T,\cdot)}{\Longrightarrow} H$, where φ flow for some T



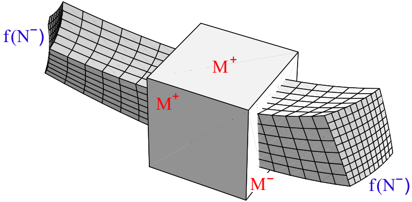
- check covering $N_4 \overset{\varphi(T,\cdot)}{\Longrightarrow} H$ for some T

2D-2D transversal connections

Idea:

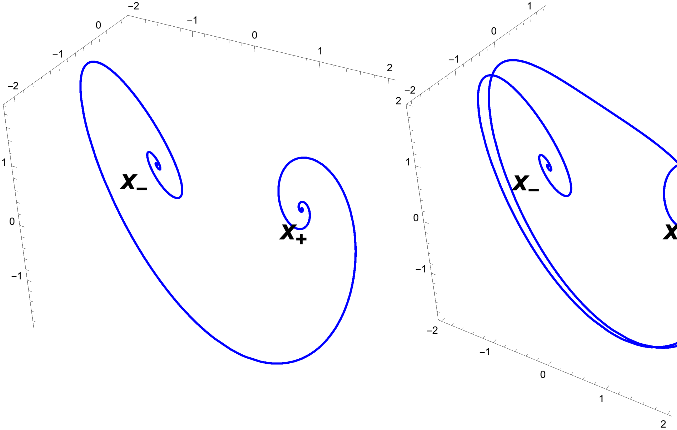
$$\{y=0\}\cap\{N_1,N_2,N_4\} \stackrel{P}{\Longrightarrow} \cdots \text{follow 4PO} \cdots \stackrel{P}{\Longrightarrow} N_4 \stackrel{P}{\Longrightarrow} H \stackrel{P}{\Longrightarrow} H$$

- slice of $y=0$ is horizontal disc in N_1, N_2, N_4
- transport it following four periodic orbit for some finite time and stop in N_4
- construct H – 3D set centered at equilibrium and spanned on eigenvectors
- check $H \stackrel{\varphi(T,\cdot)}{\Longrightarrow} H$, where φ flow for some T



- check covering $N_4 \stackrel{\varphi(T,\cdot)}{\Longrightarrow} H$ for some T

Transversal heteroclinic connections



Computational aspects:

- estimate stable manifold near equilibrium

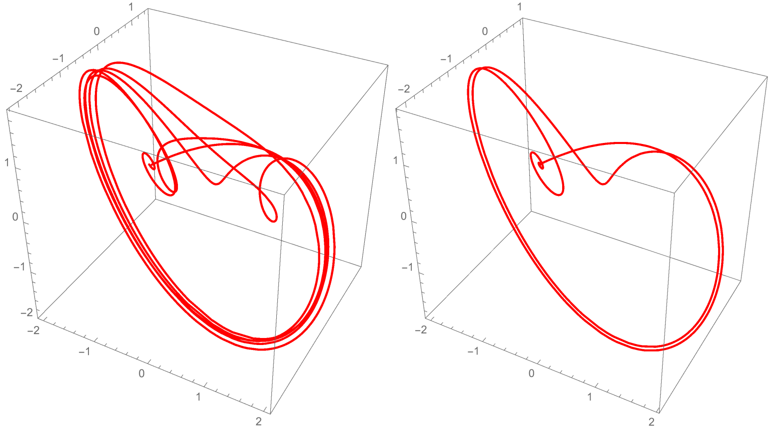
  (there is an explicit Lyapunov function in the half-plane)

- check covering between sets of different dimension

  $N_4 \Longrightarrow H \Longrightarrow H$

Idea:

$$\text{manifold} \xrightarrow{W} N_4 \xrightarrow{P} \cdots \text{follow 4PO} \cdots \xrightarrow{P} N_4 \xrightarrow{P} H \xrightarrow{P} H$$



This time the C++ program is to long to show it:)