Computer-assisted proofs in dynamics
Part II: smooth methods

Daniel Wilczak

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

June 19, 2016

Outline of Part II:

1. $\mathcal{C}^1$-algorithms for ODEs
2. Derivatives of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study:
   - hyperbolic periodic orbits
   - branches of periodic orbits
5. Some "good" abstract theorem and its application:
   - hyperbolic horseshoe in the Rössler system
   - heteroclinic connections between periodic orbits
   - hyperbolic attractor in the Kuznetsov system
6. $\mathcal{C}^r$-algorithm for ODEs and Poincaré maps
7. CAPD library: derivatives of flows and Poincaré maps
8. Case study:
   - bifurcations of Halo orbits in the CR3BP
   - stability of elliptic solutions: invariant tori

Outline of Part II:

1. $\mathcal{C}^1$-algorithms for ODEs
2. Derivatives of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study:
   - hyperbolic periodic orbits
   - branches of periodic orbits
5. Some “good” abstract theorem and its application:
   - hyperbolic horseshoe in the Rössler system
   - heteroclinic connections between periodic orbits
   - hyperbolic attractor in the Kuznetsov system
6. $\mathcal{C}^r$-algorithm for ODEs and Poincaré maps
7. CAPD library: derivatives of flows and Poincaré maps
8. Case study:
   - bifurcations of Halo orbits in the CR3BP
   - stability of elliptic solutions: invariant tori

Outline of Part II:

1. $\mathcal{C}^1$-algorithms for ODEs
2. Derivatives of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study:
   - hyperbolic periodic orbits
   - branches of periodic orbits
5. Some “good” abstract theorem and its application:
   - hyperbolic horseshoe in the Rössler system
   - heteroclinic connections between periodic orbits
   - hyperbolic attractor in the Kuznetsov system
6. $\mathcal{C}^r$-algorithm for ODEs and Poincaré maps
7. CAPD library: derivatives of flows and Poincaré maps
8. Case study:
   - bifurcations of Halo orbits in the CR3BP
   - stability of elliptic solutions: invariant tori

Outline of Part II:

1. $\mathcal{C}^1$-algorithms for ODEs
2. Derivatives of Poincaré maps
3. CAPD library: ODE solvers and Poincaré maps
4. Case study:
   - hyperbolic periodic orbits
   - branches of periodic orbits
5. Some “good” abstract theorem and its application:
   - hyperbolic horseshoe in the Rössler system
   - heteroclinic connections between periodic orbits
   - hyperbolic attractor in the Kuznetsov system
6. $\mathcal{C}^r$-algorithm for ODEs and Poincaré maps
7. CAPD library: derivatives of flows and Poincaré maps
8. Case study:
   - bifurcations of Halo orbits in the CR3BP
   - stability of elliptic solutions: invariant tori

$\mathcal{C}^1$ Solvers

Problem to solve:

$$
\begin{aligned}
x'(t) = f(x(t)) \quad &- \quad \text{main ODE} \\
V'(t) = Df(x(t)) \cdot V(t) \quad &- \quad \text{variational equation}
\end{aligned}
$$

or:

$$V(t) \;=\; D_x\varphi(t, x)$$

Initial conditions:

$$
\begin{aligned}
x(0) \;&\in\; [X] \\
V(0) \;&\in\; [V] \qquad \text{often } [V] = \{\mathrm{Id}\}
\end{aligned}
$$

Problem to solve:

$$\begin{aligned} x'(t) &= f(x(t)) \quad - \quad \text{main ODE} \\ V'(t) &= Df(x(t)) \cdot V(t) \quad - \quad \text{variational equation} \end{aligned}$$

or:

$$V(t) \;=\; D_x\varphi(t, x)$$

Initial conditions:

$$\begin{aligned} x(0) &\in [X] \\ V(0) &\in [V] \qquad \text{often } [V] = \{\mathrm{Id}\} \end{aligned}$$

Problem to solve:

$$
\begin{aligned}
x'(t) &= f(x(t)) \quad - \quad \text{main ODE} \\
V'(t) &= Df(x(t)) \cdot V(t) \quad - \quad \text{variational equation}
\end{aligned}
$$

or:

$$
V(t) \;=\; D_x\varphi(t,x)
$$

Initial conditions:

$$
\begin{aligned}
x(0) &\in [X] \\
V(0) &\in [V] \qquad \text{often } [V] = \{\mathrm{Id}\}
\end{aligned}
$$

Example (van der Pol oscillator)

$$x' = y, \quad y' = (1 - x^2)y - x$$

Full $\mathcal{C}^1$ system

$$\begin{cases} x' &= y \\ y' &= (1 - x^2)y - x \\ V'_{11} &= V_{21} \\ V'_{12} &= V_{22} \\ V'_{21} &= V_{21}(1 - x^2) - V_{11}(1 + 2xy) \\ V'_{22} &= V_{22}(1 - x^2) - V_{12}(1 + 2xy) \end{cases}$$

One can apply any $\mathcal{C}^0$ solver to the above system but ...

Complexity problem

Solvers (at least VNODE, CAPD) are $O\left(\text{dimension}^3\right)$.

(to reduce wrapping effect)

For variational equations this gives $O((\text{dimension}^2)^3)$.

Example (van der Pol oscillator)

$$x' = y, \quad y' = (1 - x^2)y - x$$

Full $\mathcal{C}^1$ system

$$\begin{cases} x' &= y \\ y' &= (1 - x^2)y - x \\ V'_{11} &= V_{21} \\ V'_{12} &= V_{22} \\ V'_{21} &= V_{21}(1 - x^2) - V_{11}(1 + 2xy) \\ V'_{22} &= V_{22}(1 - x^2) - V_{12}(1 + 2xy) \end{cases}$$

One can apply any $\mathcal{C}^0$ solver to the above system but ...

Complexity problem

Solvers (at least VNODE, CAPD) are $O\left(\text{dimension}^3\right)$.

(to reduce wrapping effect)

For variational equations this gives $O((\text{dimension}^2)^3)$.

Example (van der Pol oscillator)

$$x' = y, \quad y' = (1 - x^2)y - x$$

Full $\mathcal{C}^1$ system

$$\begin{cases} x' &= y \\ y' &= (1 - x^2)y - x \\ V'_{11} &= V_{21} \\ V'_{12} &= V_{22} \\ V'_{21} &= V_{21}(1 - x^2) - V_{11}(1 + 2xy) \\ V'_{22} &= V_{22}(1 - x^2) - V_{12}(1 + 2xy) \end{cases}$$

One can apply any $\mathcal{C}^0$ solver to the above system but ...

Complexity problem

Solvers (at least VNODE, CAPD) are $O\left(\text{dimension}^3\right)$.

(to reduce wrapping effect)

For variational equations this gives $O((\text{dimension}^2)^3)$.

Example (van der Pol oscillator)

$$x' = y, \quad y' = (1 - x^2)y - x$$

Full $\mathcal{C}^1$ system

$$\begin{cases} x' &= y \\ y' &= (1 - x^2)y - x \\ V'_{11} &= V_{21} \\ V'_{12} &= V_{22} \\ V'_{21} &= V_{21}(1 - x^2) - V_{11}(1 + 2xy) \\ V'_{22} &= V_{22}(1 - x^2) - V_{12}(1 + 2xy) \end{cases}$$

One can apply any $\mathcal{C}^0$ solver to the above system but ...

Complexity problem

Solvers (at least VNODE, CAPD) are $O\left(\text{dimension}^3\right)$.

(to reduce wrapping effect)

For variational equations this gives $O((\text{dimension}^2)^3)$.

Fact:

$$V'(t) = Df(x(t)) \cdot V(t)$$

is nonautonomous linear in V. Thus

$$D_x\varphi(t+h,x) = D_x\varphi(h,\varphi(t,x))D_x\varphi(t,x)$$

It is enough to compute

$$D_x\varphi(h,[X])$$

where h is the time step.

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

Structure of variational equation

Fact:

$$V'(t) = Df(x(t)) \cdot V(t)$$

is nonautonomous linear in V. Thus

$$D_x\varphi(t+h, x) = D_x\varphi(h, \varphi(t, x))D_x\varphi(t, x)$$

It is enough to compute

$$D_x\varphi(h, [X])$$

where h is the time step.

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

Structure of variational equation

Fact:

$$V'(t) = Df(x(t)) \cdot V(t)$$

is nonautonomous linear in V. Thus

$$D_x\varphi(t+h, x) = D_x\varphi(h, \varphi(t, x)) D_x\varphi(t, x)$$

It is enough to compute

$$D_x\varphi(h, [X])$$

where h is the time step.

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

Structure of variational equation

Φ – numerical method for ODE

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

Wrapping effect reduced by

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

Key observation:

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

$D_x\Phi(h,[X])$ – is already computed in $\mathcal{C}^0$ step.

Additional cost:

- rough enclosure for variational part
- remainder R_V for variational part
- propagation of $V(t)$ (wrapping effect reduction for V)

Structure of variational equation

Φ – numerical method for ODE

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

Wrapping effect reduced by

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

Key observation:

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

$D_x\Phi(h,[X])$ – is already computed in $\mathcal{C}^0$ step.

Additional cost:

- rough enclosure for variational part
- remainder R_V for variational part
- propagation of $V(t)$ (wrapping effect reduction for V)

Structure of variational equation

Φ – numerical method for ODE

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

Wrapping effect reduced by

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

Key observation:

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

$D_x\Phi(h,[X])$ – is already computed in $\mathcal{C}^0$ step.

Additional cost:

- rough enclosure for variational part
- remainder R_V for variational part
- propagation of $V(t)$ (wrapping effect reduction for V)

Structure of variational equation

Φ – numerical method for ODE

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

Wrapping effect reduced by

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

Key observation:

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

$D_x\Phi(h, [X])$ – is already computed in $\mathcal{C}^0$ step.

Additional cost:

- rough enclosure for variational part
- remainder R_V for variational part
- propagation of $V(t)$ (wrapping effect reduction for V)

Rough enclosure for variational equation

Fact:
If $\varphi(h,[X])$ exists then $D_x\varphi(h,[X])$ does, too.

Strategy:
$[Y]$ – enclosure for $\mathcal{C}^0$ part

- FOE: find $[W]$ such that

$$\mathrm{Id} + [0,h]Df([Y])\cdot[W] \subset [W]$$

- HOE: define

$$[W] = \sum_{i=0}^{r}[V]^{[r]}[0,h]^r + [R_V]$$

  and check

$$[W]^{[r+1]}[0,h]^{r+1} \subset [R_V]$$

Rough enclosure for variational equation

Fact:
If $\varphi(h,[X])$ exists then $D_x\varphi(h,[X])$ does, too.

Strategy:
$[Y]$ – enclosure for $\mathcal{C}^0$ part

- FOE: find $[W]$ such that
$$\mathrm{Id} + [0,h]Df([Y])\cdot[W] \subset [W]$$
- HOE: define
$$[W] = \sum_{i=0}^{r}[V]^{[r]}[0,h]^r + [R_V]$$
and check
$$[W]^{[r+1]}[0,h]^{r+1} \subset [R_V]$$

Rough enclosure for variational equation

Fact:

If $\varphi(h,[X])$ exists then $D_x\varphi(h,[X])$ does, too.

Strategy:

$[Y]$ – enclosure for $\mathcal{C}^0$ part

- FOE: find $[W]$ such that

$$\mathrm{Id} + [0,h]Df([Y])\cdot[W] \subset [W]$$

- HOE: define

$$[W] = \sum_{i=0}^{r} [V]^{[r]}[0,h]^r + [R_V]$$

and check

$$[W]^{[r+1]}[0,h]^{r+1} \subset [R_V]$$

Rough enclosure for variational equation

Fact:
If $\varphi(h,[X])$ exists then $D_x\varphi(h,[X])$ does, too.

Strategy:
$[Y]$ – enclosure for $\mathcal{C}^0$ part

- FOE: find $[W]$ such that
$$\mathrm{Id} + [0,h]Df([Y])\cdot[W] \subset [W]$$
- HOE: define
$$[W] = \sum_{i=0}^{r}[V]^{[r]}[0,h]^r + [R_V]$$
and check
$$[W]^{[r+1]}[0,h]^{r+1} \subset [R_V]$$

Propagation of matrices

Input:

$$\begin{aligned} D_x\varphi(h,\varphi(t,x)) &\in [A] \\ D_x\varphi(t,x) &\in X_0 + C[R_0] + B[R] \end{aligned}$$

Then

$$D_x\varphi(t+h,x) \in [A]X_0 + ([A]C)[R_0] + ([A]B)[R]$$

and use QR-like strategies to propagate products.

Facts

- worse control of the wrapping effect
  (dependency of V wrt to V is not used)
- much faster than direct application of a C^0 algorithm

Propagation of matrices

Input:

$$\begin{aligned} D_x\varphi(h,\varphi(t,x)) &\in [A] \\ D_x\varphi(t,x) &\in X_0 + C[R_0] + B[R] \end{aligned}$$

Then

$$D_x\varphi(t+h,x) \in [A]X_0 + ([A]C)[R_0] + ([A]B)[R]$$

and use QR-like strategies to propagate products.

Facts

- worse control of the wrapping effect
  (dependency of V wrt to V is not used)
- much faster than direct application of a C^0 algorithm

Propagation of matrices

Input:

$$\begin{aligned} D_x\varphi(h,\varphi(t,x)) &\in [A] \\ D_x\varphi(t,x) &\in X_0 + C[R_0] + B[R] \end{aligned}$$

Then

$$D_x\varphi(t+h,x) \in [A]X_0 + ([A]C)[R_0] + ([A]B)[R]$$

and use QR-like strategies to propagate products.

Facts

- worse control of the wrapping effect
  (dependency of V wrt to V is not used)
- much faster than direct application of a $\mathcal{C}^0$ algorithm

Numerical methods:

We have to enclose

$$D_x\varphi(h,\varphi(t,x)) \quad \in \quad [A]$$

Taylor method

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

$$[A] = \sum_{i=0}^{r} [V]^{[r]} h^r + [R_V]$$

Hermite-Obreshkov method

I. Walawska, DW, An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs, Applied Mathematics and Computation, to appear 2016

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

Solve for $V(h) \in [A]$ using interval Krawczyk method

(The same matrices as for $\mathcal{C}^0$ part – no extra cost!)

$$\Psi_{q,p}(-h,V(h)) - \Psi_{p,q}(h,V(0)) \in [R_{VHO}]$$

Numerical methods:

We have to enclose

$$D_x\varphi(h,\varphi(t,x)) \quad \in \quad [A]$$

Taylor method

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

$$[A] = \sum_{i=0}^{r} [V]^{[r]} h^r + [R_V]$$

Hermite-Obreshkov method

I. Walawska, DW, An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs, Applied Mathematics and Computation, to appear 2016

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

Solve for $V(h) \in [A]$ using interval Krawczyk method

(The same matrices as for $\mathcal{C}^0$ part – no extra cost!)

$$\Psi_{q,p}(-h,V(h)) - \Psi_{p,q}(h,V(0)) \in [R_{VHO}]$$

Numerical methods:

We have to enclose

$$D_x\varphi(h,\varphi(t,x)) \quad \in \quad [A]$$

Taylor method

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

$$[A] = \sum_{i=0}^{r}[V]^{[r]}h^r + [R_V]$$

Hermite-Obreshkov method

I. Walawska, DW, An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs, Applied Mathematics and Computation, to appear 2016

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

Solve for $V(h) \in [A]$ using interval Krawczyk method

(The same matrices as for $\mathcal{C}^0$ part – no extra cost!)

$$\Psi_{q,p}(-h,V(h)) - \Psi_{p,q}(h,V(0)) \in [R_{VHO}]$$

We have to enclose

$$D_x\varphi(h,\varphi(t,x)) \quad \in \quad [A]$$

Taylor method

P. Zgliczyński, $\mathcal{C}^1$-Lohner algorithm, Found. Comp. Math, (2002), 2:429-465

$$[A] = \sum_{i=0}^{r} [V]^{[r]} h^r + [R_V]$$

Hermite-Obreshkov method

I. Walawska, DW, An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs, Applied Mathematics and Computation, to appear 2016

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

Solve for $V(h) \in [A]$ using interval Krawczyk method

(The same matrices as for $\mathcal{C}^0$ part – no extra cost!)

$$\Psi_{q,p}(-h,V(h)) - \Psi_{p,q}(h,V(0)) \in [R_{VHO}]$$

Derivatives of Poincaré maps:

Section:

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

Return time:

$$T : \mathbb{R}^n \to \mathbb{R}$$

Then

$$\alpha(\varphi(T(x), x)) \equiv 0$$

$$\Downarrow$$

$$\sum_{i=1}^{n} \frac{\partial\alpha}{\partial x_i}(P(x)) \left(f_i(P(x)) \frac{\partial T}{\partial x_j}(x) + \frac{\partial\varphi_i}{\partial x_j}(T(x), x) \right) = 0$$

$$\Downarrow$$

$$\frac{\partial T(x)}{\partial x_j} = -\frac{\langle \nabla\alpha(P(x))\,;\, D_{x_j}\varphi(T(x), x)\rangle}{\langle \nabla\alpha(P(x))\,;\, f(\varphi(T(x), x))\rangle}$$

Derivatives of Poincaré maps:

Section:

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

Return time:

$$T : \mathbb{R}^n \to \mathbb{R}$$

Then

$$\alpha(\varphi(T(x), x)) \equiv 0$$

$$\Downarrow$$

$$\sum_{i=1}^{n} \frac{\partial\alpha}{\partial x_i}(P(x)) \left(f_i(P(x)) \frac{\partial T}{\partial x_j}(x) + \frac{\partial\varphi_i}{\partial x_j}(T(x), x) \right) = 0$$

$$\Downarrow$$

$$\frac{\partial T(x)}{\partial x_j} = -\frac{\langle \nabla\alpha(P(x))\,;\, D_{x_j}\varphi(T(x), x)\rangle}{\langle \nabla\alpha(P(x))\,;\, f(\varphi(T(x), x))\rangle}$$

Derivatives of Poincaré maps:

Section:

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

Return time:

$$T : \mathbb{R}^n \to \mathbb{R}$$

Then

$$\alpha(\varphi(T(x), x)) \equiv 0$$

$$\Downarrow$$

$$\sum_{i=1}^{n} \frac{\partial \alpha}{\partial x_i}(P(x)) \left(f_i(P(x)) \frac{\partial T}{\partial x_j}(x) + \frac{\partial \varphi_i}{\partial x_j}(T(x), x) \right) = 0$$

$$\Downarrow$$

$$\frac{\partial T(x)}{\partial x_j} = -\frac{\langle \nabla\alpha(P(x))\,;\, D_{x_j}\varphi(T(x), x)\rangle}{\langle \nabla\alpha(P(x))\,;\, f(\varphi(T(x), x))\rangle}$$

Derivatives of Poincaré maps:

Section:

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

Return time:

$$T : \mathbb{R}^n \to \mathbb{R}$$

Then

$$\alpha(\varphi(T(x), x)) \equiv 0$$
$$\Downarrow$$
$$\sum_{i=1}^{n} \frac{\partial\alpha}{\partial x_i}(P(x)) \left(f_i(P(x)) \frac{\partial T}{\partial x_j}(x) + \frac{\partial \varphi_i}{\partial x_j}(T(x), x) \right) = 0$$
$$\Downarrow$$
$$\frac{\partial T(x)}{\partial x_j} = -\frac{\langle \nabla\alpha(P(x))\,;\, D_{x_j}\varphi(T(x), x)\rangle}{\langle \nabla\alpha(P(x))\,;\, f(\varphi(T(x), x))\rangle}$$

Derivatives of Poincaré maps:

Section:

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

Return time:

$$T : \mathbb{R}^n \to \mathbb{R}$$

Then

$$\alpha(\varphi(T(x), x)) \equiv 0$$
$$\Downarrow$$
$$\sum_{i=1}^{n} \frac{\partial \alpha}{\partial x_i}(P(x)) \left(f_i(P(x)) \frac{\partial T}{\partial x_j}(x) + \frac{\partial \varphi_i}{\partial x_j}(T(x), x) \right) = 0$$
$$\Downarrow$$
$$\frac{\partial T(x)}{\partial x_j} = -\frac{\left\langle \nabla\alpha(P(x))\,;\ D_{x_j}\varphi(T(x), x)\right\rangle}{\langle \nabla\alpha(P(x))\,;\ f(\varphi(T(x), x))\rangle}$$

Derivatives of Poincaré maps:

Given bounds for:

- $T([X])$ – from $\mathcal{C}^0$ part
- $P([X])$ – from $\mathcal{C}^0$ part
- $D_x\varphi(T([X]),[X])$

Compute:

- derivatives for return time

$$\frac{\partial T}{\partial x_j}(x) = -\frac{\langle \nabla\alpha(P(x))\,;\, D_{x_j}\varphi(T(x),x)\rangle}{\langle \nabla\alpha(P(x))\,;\, f(\varphi(T(x),x))\rangle}$$

- derivatives of Poincaré map

$$\frac{\partial P_i}{\partial x_j}(x) = \frac{\partial \varphi_i}{\partial x_j}(T(x),x) + \sum_{k=1}^{n} f_i(P(x))\frac{\partial T}{\partial x_j}(x)$$

Derivatives of Poincaré maps:

Given bounds for:

- $T([X])$ – from $\mathcal{C}^0$ part
- $P([X])$ – from $\mathcal{C}^0$ part
- $D_x\varphi(T([X]),[X])$

Compute:

- derivatives for return time

$$\frac{\partial T}{\partial x_j}(x) = -\frac{\langle \nabla\alpha(P(x))\,;\,D_{x_j}\varphi(T(x),x)\rangle}{\langle \nabla\alpha(P(x))\,;\,f(\varphi(T(x),x))\rangle}$$

- derivatives of Poincaré map

$$\frac{\partial P_i}{\partial x_j}(x) = \frac{\partial \varphi_i}{\partial x_j}(T(x),x) + \sum_{k=1}^{n} f_i(P(x))\frac{\partial T}{\partial x_j}(x)$$

Derivatives of Poincaré maps:

Given bounds for:

- $T([X])$ – from $\mathcal{C}^0$ part
- $P([X])$ – from $\mathcal{C}^0$ part
- $D_x\varphi(T([X]),[X])$

Compute:

- derivatives for return time

$$\frac{\partial T}{\partial x_j}(x) = -\frac{\langle \nabla\alpha(P(x))\,;\, D_{x_j}\varphi(T(x),x)\rangle}{\langle \nabla\alpha(P(x))\,;\, f(\varphi(T(x),x))\rangle}$$

- derivatives of Poincaré map

$$\frac{\partial P_i}{\partial x_j}(x) = \frac{\partial \varphi_i}{\partial x_j}(T(x),x) + \sum_{k=1}^{n} f_i(P(x))\frac{\partial T}{\partial x_j}(x)$$

Derivatives of Poincaré maps:

Given bounds for:

- $T([X])$ – from $\mathcal{C}^0$ part
- $P([X])$ – from $\mathcal{C}^0$ part
- $D_x\varphi(T([X]),[X])$

Compute:

- derivatives for return time

$$\frac{\partial T}{\partial x_j}(x) = -\frac{\left\langle \nabla\alpha(P(x))\,;\ D_{x_j}\varphi(T(x),x)\right\rangle}{\left\langle \nabla\alpha(P(x))\,;\ f(\varphi(T(x),x))\right\rangle}$$

- derivatives of Poincaré map

$$\frac{\partial P_i}{\partial x_j}(x) = \frac{\partial \varphi_i}{\partial x_j}(T(x),x) + \sum_{k=1}^{n} f_i(P(x))\frac{\partial T}{\partial x_j}(x)$$

Derivatives of Poincaré maps:

Given bounds for:

- $T([X])$ – from $\mathcal{C}^0$ part
- $P([X])$ – from $\mathcal{C}^0$ part
- $D_x\varphi(T([X]),[X])$

Compute:

- derivatives for return time

$$\frac{\partial T}{\partial x_j}(x) = -\frac{\left\langle \nabla\alpha(P(x))\,;\ D_{x_j}\varphi(T(x),x)\right\rangle}{\left\langle \nabla\alpha(P(x))\,;\ f(\varphi(T(x),x))\right\rangle}$$

- derivatives of Poincaré map

$$\frac{\partial P_i}{\partial x_j}(x) = \frac{\partial \varphi_i}{\partial x_j}(T(x),x) + \sum_{k=1}^{n} f_i(P(x))\frac{\partial T}{\partial x_j}(x)$$

```cpp
/** Example of integration of variational equations **/
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
int main(){
  IMap lorenz("var:x,y,z;fun:10*(y-x),x*(28-z)-y,x*y-8*z/3;");
  IOdeSolver solver(lorenz,20);  // ODE integrator
  ITimeMap tm(solver);           // class for long time integration

  IVector u({1,5,23});
  // representation of initial condition,
  // initial condition for variational equations is Id by default
  C1HORect2Set set(u);
  // integrate until T=2 and print result
  cout << "phi(2,u)=" << tm(2.,set) << endl;
  // print monodromy matrix
  cout << "D_x phi(t,u) = " << (IMatrix)set << endl;

  ITimeMap::SolutionCurve solution(0.);
  C1HORect2Set s(u);
  // integrate and record trajectory
  tm(2.,s,solution);
  cout << "solution(1) = " << solution(1) << endl;
  cout << "D_x solution(1.5) = " << solution.derivative(1.5) << endl;
  return 0;
}
```

```
/* Output of the program:
phi(2,u)={[3.57516, 3.57516],[-0.797643, -0.797643],[27.9102, 27.9102]
D_x phi(t,u) = {
{[-1.95841, -1.95841],[-3.28406, -3.28406],[-0.614346, -0.614346]},
{[1.17314, 1.17314],[1.71972, 1.71972],[0.813136, 0.813136]},
{[-4.75217, -4.75217],[-7.68436, -7.68436],[-2.00251, -2.00251]}
}
solution(1) = {[11.8084, 11.8084],[16.4144, 16.4144],[25.3, 25.3]}
D_x solution(1.5) = {
{[0.974059, 0.974059],[1.87338, 1.87338],[-0.125996, -0.125996]},
{[-0.276187, -0.276187],[-0.249576, -0.249576],[-0.470697, -0.470697]}
{[3.59364, 3.59364],[6.48145, 6.48145],[0.30862, 0.30862]}
}*/
```

```
/** Example of computation of derivative of Poincare map **/
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
int main(){
  cout.precision(5);
  IMap lorenz("var:x,y,z;fun:10*(y-x),x*(28-z)-y,x*y-8*z/3;");
  IOdeSolver solver(lorenz,20);         // ODE integrator
  ICoordinateSection section(3,2,27.); // section is z=27
  IPoincareMap pm(solver,section);

  C1HORect2Set set(IVector({1,5,27}));
  IMatrix Dphi(3,3);
  IVector P = pm(set,Dphi);
  // recompute derivative of flow to derivative of Poincare map
  IMatrix DP = pm.computeDP(P,Dphi);
  cout << "P(1,5,27)=" << P << endl;
  cout << "DP(1,5,27)=" << DP << endl;
}
/* Output:
P(1,5,27)={[11.361, 11.361],[14.338, 14.338],[27, 27]}
DP(1,5,27)={
{[-0.19415, -0.19415],[-0.46759, -0.46759],[-0.087997, -0.087997]},
{[-0.4121, -0.4121],[-0.96593, -0.96593],[-0.18836, -0.18836]},
{[-1.37e-13, 1.3706e-13],[-5.2713e-13, 5.2758e-13],[-2.5335e-13, 2.533
}
*/
```

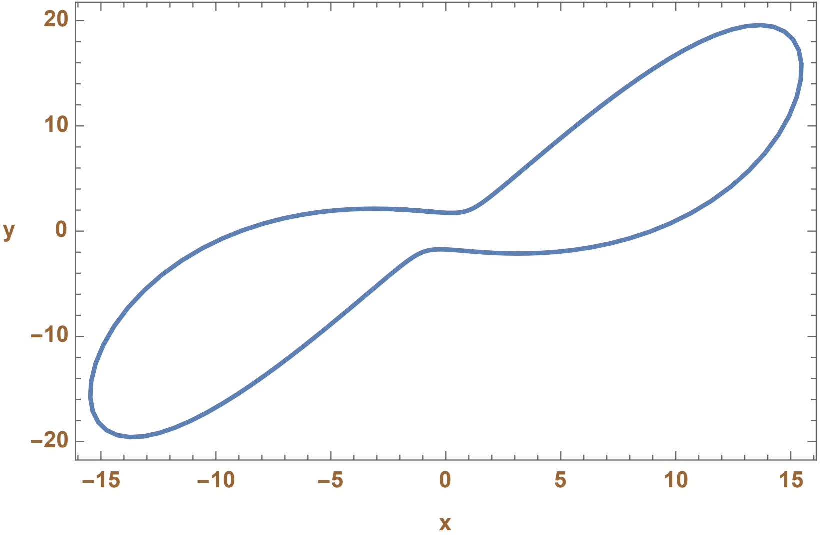
Case study

- hyperbolic periodic orbits
- branches of periodic orbits

Example (Toy example: Lorenz system)

$$x' = 10(y - x), \qquad y' = x(28 - z) - y, \qquad z' = xy - 8z/3$$

Goal: prove that there is a hyperbolic periodic orbit



Methodology:

- Poincaré map

$$\Pi = \{(x, y, 27) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^4(u) - u$$

has zero

Data:

$$u_0 = (-2.14737, 2.07805)$$
$$[r] = ([-10^{-5}, 10^{-5}], [-10^{-5}, 10^{-5}])$$

Check

$$[N] := -\left(P^4(u_0 + [r]) - \mathrm{Id}\right)^{-1} \cdot \left(P^4(u_0) - u_0\right) \subset [r]$$

Compute eigenvalues of $DP^4(u_0 + [r])$

(Could be improved to $P^4(u_0 + [N])$)

Methodology:

- Poincaré map

$$\Pi = \{(x, y, 27) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^4(u) - u$$

has zero

Data:

$$u_0 = (-2.14737, 2.07805)$$

$$[r] = ([-10^{-5}, 10^{-5}], [-10^{-5}, 10^{-5}])$$

Check

$$[N] := -\left(P^4(u_0 + [r]) - \mathrm{Id}\right)^{-1} \cdot \left(P^4(u_0) - u_0\right) \subset [r]$$

Compute eigenvalues of $DP^4(u_0 + [r])$

(Could be improved to $P^4(u_0 + [N])$)

Methodology:

- Poincaré map

$$\Pi = \{(x, y, 27) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^4(u) - u$$

has zero

Data:

$$u_0 = (-2.14737, 2.07805)$$
$$[r] = ([-10^{-5}, 10^{-5}], [-10^{-5}, 10^{-5}])$$

Check

$$[N] := -\left(P^4(u_0 + [r]) - \mathrm{Id}\right)^{-1} \cdot \left(P^4(u_0) - u_0\right) \subset [r]$$

Compute eigenvalues of $DP^4(u_0 + [r])$

(Could be improved to $P^4(u_0 + [N])$)

Methodology:

- Poincaré map

$$\Pi = \{(x, y, 27) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^4(u) - u$$

has zero

Data:

$$u_0 = (-2.14737, 2.07805)$$
$$[r] = ([-10^{-5}, 10^{-5}], [-10^{-5}, 10^{-5}])$$

Check

$$[N] := -\left(P^4(u_0 + [r]) - \mathrm{Id}\right)^{-1} \cdot \left(P^4(u_0) - u_0\right) \subset [r]$$

Compute eigenvalues of $DP^4(u_0 + [r])$

(Could be improved to $P^4(u_0 + [N])$)

Methodology:

- Poincaré map

$$\Pi = \{(x, y, 27) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^4(u) - u$$

has zero

Data:

$$u_0 = (-2.14737, 2.07805)$$
$$[r] = ([-10^{-5}, 10^{-5}], [-10^{-5}, 10^{-5}])$$

Check

$$[N] := -\left(P^4(u_0 + [r]) - \mathrm{Id}\right)^{-1} \cdot \left(P^4(u_0) - u_0\right) \subset [r]$$

Compute eigenvalues of $DP^4(u_0 + [r])$

(Could be improved to $P^4(u_0 + [N])$)

```
int main(){
  IMap lorenz("var:x,y,z;fun:10*(y-x),x*(28-z)-y,x*y-8*z/3;");
  IOdeSolver solver(lorenz,20);        // ODE integrator
  ICoordinateSection section(3,2,27.); // section is z=27
  IPoincareMap pm(solver,section);
  // very rough approximation of a periodic point
  IVector u0({-2.14737, 2.07805, 27});
  IVector r = IVector({1.,1.,0.})*interval(-1e-5,1e-5);
  C0HOTripletonSet s0(u0);
  // compute fu0:=P^4(u0)-u0 and project it onto (x,y)
  IVector fu0( 2, (pm(s0,4) - u0).begin() );
  // compute derivative
  C1HORect2Set s1(u0+r);
  IMatrix Dphi(3,3);
  IVector Pu = pm(s1,Dphi,4);
  IMatrix DP = pm.computeDP(Pu,Dphi);
  // projection of DP^4(u0+r)-Id onto 2D subspace
  IMatrix M({{DP(1,1)-1,DP(1,2)},{DP(2,1),DP(2,2)-1}});
  // enclose -(DP^4(u0+r)-Id)^{-1}*(P^4(u0)-u0)
  IVector N = - matrixAlgorithms::gauss(M,fu0);
  cout << "validated? " << subset(N,IVector(2,r.begin())) << endl;
  cout << "DP=" << IMatrix({{DP(1,1),DP(1,2)},{DP(2,1),DP(2,2)}});
  // explicit formula for eigenvalues of a 2x2 matrix
  interval t = sqrt(4*DP(2,1)*DP(1,2) + sqr(DP(1,1)-DP(2,2)));
  cout << "\nlambda1=" << 0.5*(DP(1,1)+DP(2,2)-t);
  cout << "\nlambda2=" << 0.5*(DP(1,1)+DP(2,2)+t);
}
```

```
/* Output of the program:
validated? 1
DP={
{[1.01919, 1.0261],[2.72595, 2.74535]},
{[1.37796, 1.38107],[3.68603, 3.69458]}
}
lambda1=[-0.00979719, 0.00979936]
lambda2=[4.70315, 4.72275]
*/
```

Example (Rössler system)

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

Goal: localize with high accuracy three periodic orbits

Methodology:

- Poincaré map

$$\Pi = \{(0, y, z) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^n(u) - u$$

  has zero

```
/* Output of the program:
(validated?, accuracy) = true, 7.87471e-41
(validated?, accuracy) = true, 3.35732e-41
(validated?, accuracy) = true, 3.36216e-41 */
```

Remark 1: one can check that these orbits are of period 1, 2, 3.

Remark 2: accuracy – diameter of returned enclosure

Example (Rössler system)

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

Goal: localize with high accuracy three periodic orbits

Methodology:

- Poincaré map

$$\Pi = \{(0, y, z) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^n(u) - u$$

  has zero

```
/* Output of the program:
(validated?, accuracy) = true, 7.87471e-41
(validated?, accuracy) = true, 3.35732e-41
(validated?, accuracy) = true, 3.36216e-41 */
```

Remark 1: one can check that these orbits are of period 1, 2, 3.

Remark 2: accuracy – diameter of returned enclosure

Example (Rössler system)

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

Goal: localize with high accuracy three periodic orbits

Methodology:

- Poincaré map

$$\Pi = \{(0, y, z) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^n(u) - u$$

  has zero

```
/* Output of the program:
(validated?, accuracy) = true, 7.87471e-41
(validated?, accuracy) = true, 3.35732e-41
(validated?, accuracy) = true, 3.36216e-41 */
```

Remark 1: one can check that these orbits are of period 1, 2, 3.

Remark 2: accuracy – diameter of returned enclosure

Example (Rössler system)

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

Goal: localize with high accuracy three periodic orbits

Methodology:

- Poincaré map

$$\Pi = \{(0, y, z) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^n(u) - u$$

has zero

```
/* Output of the program:
(validated?, accuracy) = true, 7.87471e-41
(validated?, accuracy) = true, 3.35732e-41
(validated?, accuracy) = true, 3.36216e-41 */
```

Remark 1: one can check that these orbits are of period 1, 2, 3.

Remark 2: accuracy – diameter of returned enclosure

Example (Rössler system)

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

Goal: localize with high accuracy three periodic orbits

Methodology:

- Poincaré map

$$\Pi = \{(0, y, z) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^n(u) - u$$

  has zero

```
/* Output of the program:
(validated?, accuracy) = true, 7.87471e-41
(validated?, accuracy) = true, 3.35732e-41
(validated?, accuracy) = true, 3.36216e-41 */
```

Remark 1: one can check that these orbits are of period 1, 2, 3.

Remark 2: accuracy – diameter of returned enclosure

Example (Rössler system)

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

Goal: localize with high accuracy three periodic orbits

Methodology:

- Poincaré map

$$\Pi = \{(0, y, z) : x, y \in \mathbb{R}\}, \qquad P : \Pi \to \Pi$$

- Prove (interval Newton operator) that the function

$$f(u) := P^n(u) - u$$

has zero

```
/* Output of the program:
(validated?, accuracy) = true, 7.87471e-41
(validated?, accuracy) = true, 3.35732e-41
(validated?, accuracy) = true, 3.36216e-41 */
```

Remark 1: one can check that these orbits are of period 1, 2, 3.

Remark 2: accuracy – diameter of returned enclosure

```
void check(MpFloat y, MpFloat z, int n, double e=1e-22){
 MpIMap rossler("var:x,y,z;fun:-(y+z),x+0.2*y,0.2+z*(x-5.7);");
 MpIOdeSolver solver(rossler,50);       // ODE integrator
 MpICoordinateSection section(3,0.);    // section is x=0
 MpIPoincareMap pm(solver,section, poincare::MinusPlus);
 // approximate periodic point and a ball
 MpIVector u0({MpInterval(0.),y,z});
 MpIVector r({MpInterval(0.),MpInterval(-e,e),MpInterval(-e,e)});
 MpC0TripletonSet s0(u0);
 // compute P^n(u0)-u0 and project it onto (y,z)
 MpIVector fu0( 2, (pm(s0,n) - u0).begin() + 1 );
 MpC1Rect2Set s1(u0+r);    // compute derivative on u0+r
 MpIMatrix Dphi(3,3);
 MpIVector u = pm(s1,Dphi,n);
 MpIMatrix DP = pm.computeDP(u,Dphi);
 // projection of DP^n(u0+r)-Id onto 2D subspace
 MpIMatrix M({{DP(2,2)-1.,DP(2,3)},{DP(3,2),DP(3,3)-1.}});
 // enclose -(DP^n(u0+r)-Id)^{-1}*(P^n(u0)-u0)
 MpIVector N = - matrixAlgorithms::gauss(M,fu0);
 cout << boolalpha << "\n(validated?, accuracy) = "
      << subset(N,MpIVector(2,r.begin()+1)) << ", " << maxWidth(N);
}
int main(){
 MpFloat::setDefaultPrecision(200);
 check("-8.3809417428298764518359 3","0.029590060630667103833007 45",1);
 check("-5.4240738226652042264003 6","0.031081210807876446203326 08",2);
 check("-6.2331586285379746559602 9","0.030640111658160570583792 28",3);
}
```

Branches of Halo orbits

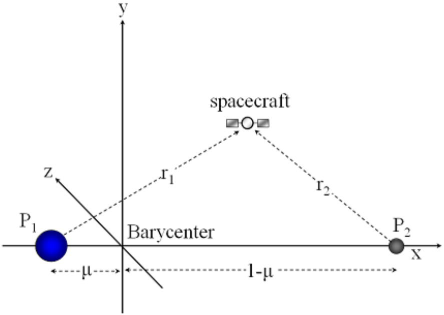
in

Circular Restricted Three Body Problem

I. Walawska, DW, Bifurcations and continuation of Halo orbits, in preparation.

Circular Restricted Three Body Problem

$$\begin{cases} \ddot{x} - 2\dot{y} = \frac{\partial \Omega(x,y,z)}{\partial x} \\ \ddot{y} + 2\dot{x} = \frac{\partial \Omega(x,y,z)}{\partial y} \\ \ddot{z} = \frac{\partial \Omega(x,y,z)}{\partial z} \end{cases}$$



where

$$\Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2}$$

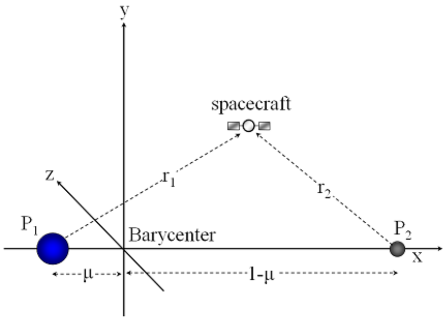
and

$$r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2},$$

$$r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}.$$

Circular Restricted Three Body Problem

$$\begin{cases} \ddot{x} - 2\dot{y} = \frac{\partial \Omega(x,y,z)}{\partial x} \\ \ddot{y} + 2\dot{x} = \frac{\partial \Omega(x,y,z)}{\partial y} \\ \ddot{z} = \frac{\partial \Omega(x,y,z)}{\partial z} \end{cases}$$



where

$$\Omega(x,y,z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2}$$

and

$$\begin{aligned} r_1 &= \sqrt{(x+\mu)^2 + y^2 + z^2}, \\ r_2 &= \sqrt{(x-1+\mu)^2 + y^2 + z^2}. \end{aligned}$$

Properties of the CR3BP

- Jacobi integral

$$C(x, y, z, \dot{x}, \dot{y}, \dot{z}) = 2\Omega(x, y, z) - (\dot{x}^2 + \dot{y}^2 + \dot{z}^2).$$

- reversing symmetry

$$R\colon (x(t), y(t), z(t)) \longrightarrow (x(-t), -y(-t), z(-t))$$

- symmetry S – reflection with respect to $z = 0$

Properties of the CR3BP

- Jacobi integral

$$C(x, y, z, \dot{x}, \dot{y}, \dot{z}) = 2\Omega(x, y, z) - (\dot{x}^2 + \dot{y}^2 + \dot{z}^2).$$

- reversing symmetry

$$R\colon\ (x(t), y(t), z(t)) \longrightarrow (x(-t), -y(-t), z(-t))$$

- symmetry S – reflection with respect to $z = 0$

Properties of the CR3BP

- Jacobi integral

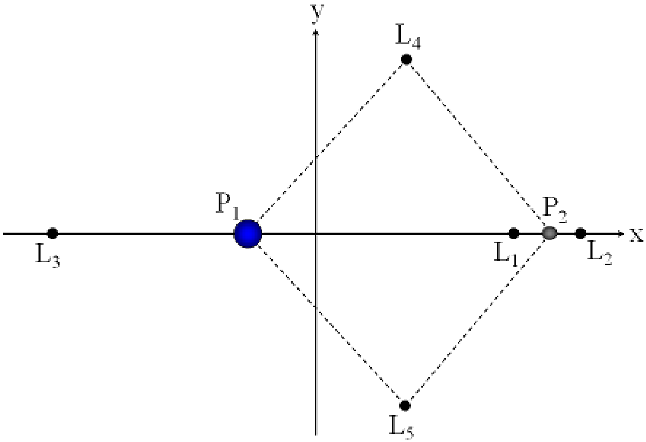
  $$C(x, y, z, \dot{x}, \dot{y}, \dot{z}) = 2\Omega(x, y, z) - (\dot{x}^2 + \dot{y}^2 + \dot{z}^2).$$

- reversing symmetry

  $$R\colon (x(t), y(t), z(t)) \longrightarrow (x(-t), -y(-t), z(-t))$$

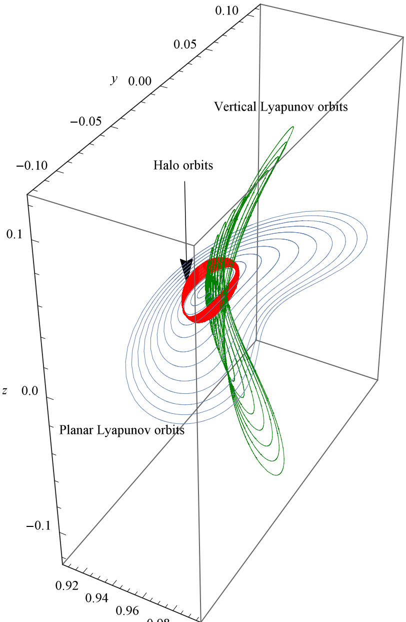
- symmetry S – reflection with respect to $z = 0$

Five libration points (all in $z = 0$ plane)



$L_{1,2}$ are of saddle $\times$ centre $\times$ centre type

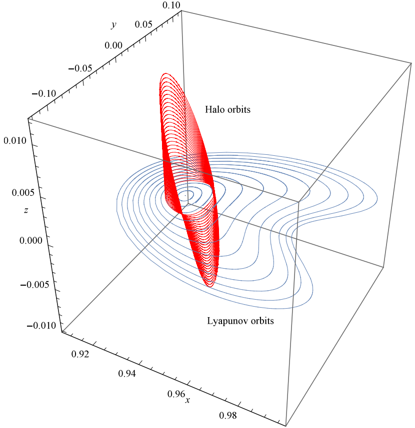
Planar and vertical Lyapunov orbits near L_1



Halo orbits

- out of z-plane R-symmetric periodic orbits which bifurcate from planar Lyapunov family
- used in space missions

Goal:
prove that there is a branch of Halo orbits

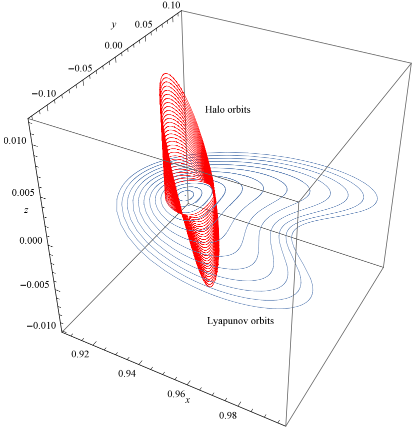


Halo orbits

- out of z-plane R-symmetric periodic orbits which bifurcate from planar Lyapunov family
- used in space missions

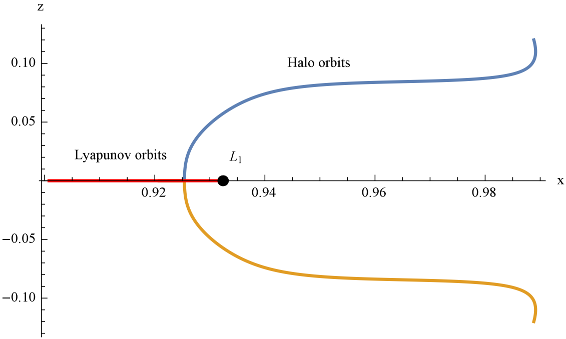
Goal:

prove that there is a branch of Halo orbits



Methodology:

- $\Pi = \{(x, y = 0, z, \dot{x}, \dot{y}, \dot{z}) \in \mathbb{R}^6\}$ – Poincaré section
- $P : \Pi \to \Pi$ – Poincaré map
- Halo orbits can be parameterized by z-amplitude



- They are R-symmetric, thus exactly twice intersect

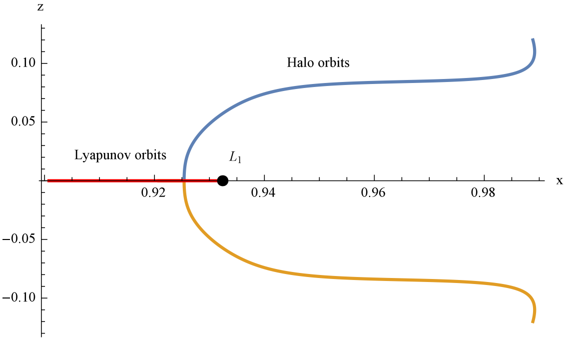
$$\mathrm{Fix}(R) = \{(x, 0, z, 0, \dot{y}, 0) \in \Pi\}$$

- Find zeros of the function $f : \mathbb{R}^3 \to \mathbb{R}^2$

$$f(z, x, \dot{y}) := \pi_{(\dot{x}, \dot{z})} P(x, 0, z, 0, \dot{y}, 0)$$

Methodology:

- $\Pi = \{(x, y = 0, z, \dot{x}, \dot{y}, \dot{z}) \in \mathbb{R}^6\}$ – Poincaré section
- $P \colon \Pi \to \Pi$ – Poincaré map
- Halo orbits can be parameterized by z-amplitude



- They are R-symmetric, thus exactly twice intersect

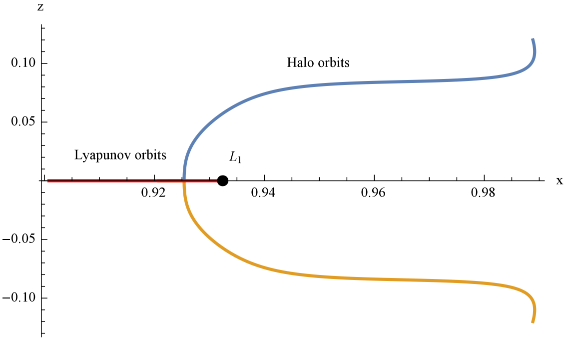
$$\mathrm{Fix}(R) = \{(x, 0, z, 0, \dot{y}, 0) \in \Pi\}$$

- Find zeros of the function $f \colon \mathbb{R}^3 \to \mathbb{R}^2$

$$f(z, x, \dot{y}) := \pi_{(\dot{x}, \dot{z})} P(x, 0, z, 0, \dot{y}, 0)$$

Methodology:

- $\Pi = \{(x, y=0, z, \dot{x}, \dot{y}, \dot{z}) \in \mathbb{R}^6\}$ – Poincaré section
- $P\colon \Pi \to \Pi$ – Poincaré map
- Halo orbits can be parameterized by z-amplitude



- They are R-symmetric, thus exactly twice intersect

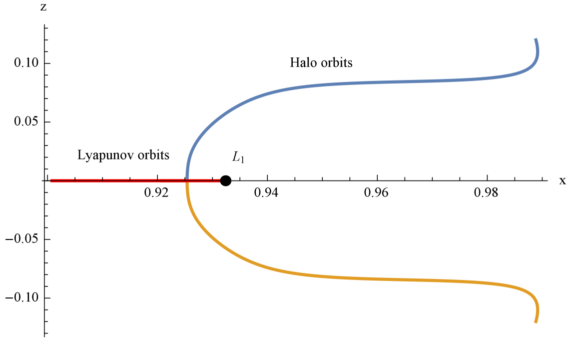
$$\mathrm{Fix}(R) = \{(x, 0, z, 0, \dot{y}, 0) \in \Pi\}$$

- Find zeros of the function $f\colon \mathbb{R}^3 \to \mathbb{R}^2$

$$f(z, x, \dot{y}) := \pi_{(\dot{x},\dot{z})} P(x, 0, z, 0, \dot{y}, 0)$$

Methodology:

- $\Pi = \{(x, y = 0, z, \dot{x}, \dot{y}, \dot{z}) \in \mathbb{R}^6\}$ – Poincaré section
- $P\colon \Pi \to \Pi$ – Poincaré map
- Halo orbits can be parameterized by z-amplitude



- They are R-symmetric, thus exactly twice intersect

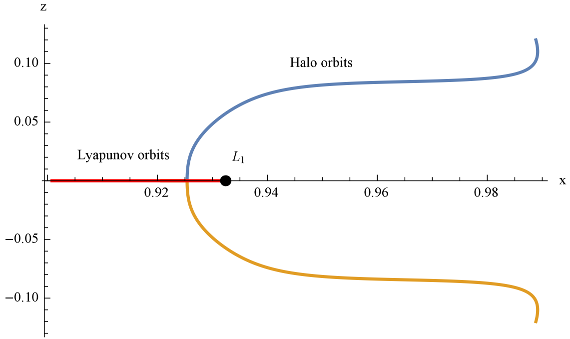
$$\mathrm{Fix}(R) = \{(x, 0, z, 0, \dot{y}, 0) \in \Pi\}$$

- Find zeros of the function $f\colon \mathbb{R}^3 \to \mathbb{R}^2$

$$f(z, x, \dot{y}) := \pi_{(\dot{x}, \dot{z})} P(x, 0, z, 0, \dot{y}, 0)$$

Methodology:

- $\Pi = \{(x, y = 0, z, \dot{x}, \dot{y}, \dot{z}) \in \mathbb{R}^6\}$ – Poincaré section
- $P \colon \Pi \to \Pi$ – Poincaré map
- Halo orbits can be parameterized by z-amplitude



- They are R-symmetric, thus exactly twice intersect

$$\mathrm{Fix}(R) = \{(x, 0, z, 0, \dot{y}, 0) \in \Pi\}$$

- Find zeros of the function $f \colon \mathbb{R}^3 \to \mathbb{R}^2$

$$f(z, x, \dot{y}) := \pi_{(\dot{x}, \dot{z})} P(x, 0, z, 0, \dot{y}, 0)$$

Interval Newton Method for parametrized functions

Lemma

Z – interval

$$W := Z \times X \times \dot{Y} := Z \times ([x_0 - \Delta x, x_0 + \Delta x] \times [\dot{y}_0 - \Delta \dot{y}, \dot{y}_0 + \Delta \dot{y}])$$

If

$$N = \begin{bmatrix} x_0 \\ \dot{y}_0 \end{bmatrix} - \left[D_{(x,\dot{y})} f(W)\right]_I^{-1} \cdot f(Z, x_0, \dot{y}_0)^T \subset \mathrm{int}(X \times \dot{Y})$$

then the solution set to

$$f(z, x, \dot{y}) = 0$$

restricted to W is a graph of a smooth function

$$Z \ni z \to (x(z), \dot{y}(z)) \in X \times \dot{Y}.$$

Proof: implicit function theorem + interval Newton method

Interval Newton Method for parametrized functions

Lemma

Z – interval

$$W := Z \times X \times \dot{Y} := Z \times ([x_0 - \Delta x, x_0 + \Delta x] \times [\dot{y}_0 - \Delta \dot{y}, \dot{y}_0 + \Delta \dot{y}])$$

If

$$N = \begin{bmatrix} x_0 \\ \dot{y}_0 \end{bmatrix} - \left[D_{(x,\dot{y})} f(W)\right]_I^{-1} \cdot f(Z, x_0, \dot{y}_0)^T \subset \operatorname{int}(X \times \dot{Y})$$

then the solution set to

$$f(z, x, \dot{y}) = 0$$

restricted to W is a graph of a smooth function

$$Z \ni z \to (x(z), \dot{y}(z)) \in X \times \dot{Y}.$$

Proof: implicit function theorem + interval Newton method

Interval Newton Method for parametrized functions

Lemma

Z – interval

$$W := Z \times X \times \dot{Y} := Z \times ([x_0 - \Delta x, x_0 + \Delta x] \times [\dot{y}_0 - \Delta \dot{y}, \dot{y}_0 + \Delta \dot{y}])$$

If

$$N = \begin{bmatrix} x_0 \\ \dot{y}_0 \end{bmatrix} - \left[D_{(x,\dot{y})} f(W)\right]_I^{-1} \cdot f(Z, x_0, \dot{y}_0)^T \subset \mathrm{int}(X \times \dot{Y})$$

then the solution set to

$$f(z, x, \dot{y}) = 0$$

restricted to W is a graph of a smooth function

$$Z \ni z \to (x(z), \dot{y}(z)) \in X \times \dot{Y}.$$

Proof: implicit function theorem + interval Newton method

Interval Newton Method for parametrized functions

Lemma

Z – interval

$$W := Z\times X\times \dot{Y} := Z\times([x_0-\Delta x, x_0+\Delta x]\times[\dot{y}_0-\Delta\dot{y}, \dot{y}_0+\Delta\dot{y}])$$

If

$$N = \begin{bmatrix} x_0 \\ \dot{y}_0 \end{bmatrix} - \left[D_{(x,\dot{y})}f(W)\right]_I^{-1}\cdot f(Z,x_0,\dot{y}_0)^T \subset \mathrm{int}(X\times\dot{Y})$$

then the solution set to

$$f(z,x,\dot{y}) = 0$$

restricted to W is a graph of a smooth function

$$Z \ni z \to (x(z),\dot{y}(z)) \in X\times\dot{Y}.$$

Proof: implicit function theorem + interval Newton method

Interval Newton Method for parametrized functions

Lemma

Z – interval

$$W := Z \times X \times \dot{Y} := Z \times ([x_0 - \Delta x, x_0 + \Delta x] \times [\dot{y}_0 - \Delta \dot{y}, \dot{y}_0 + \Delta \dot{y}])$$

If

$$N = \begin{bmatrix} x_0 \\ \dot{y}_0 \end{bmatrix} - \left[D_{(x,\dot{y})} f(W)\right]_I^{-1} \cdot f(Z, x_0, \dot{y}_0)^T \subset \mathrm{int}(X \times \dot{Y})$$

then the solution set to

$$f(z, x, \dot{y}) = 0$$

restricted to W is a graph of a smooth function

$$Z \ni z \to (x(z), \dot{y}(z)) \in X \times \dot{Y}.$$

Proof: implicit function theorem + interval Newton method

Playing with coordinate systems – flatten the curve

Consider the problem $f(z,u)=0$ and fix $z_0 \in Z$.

New coordinates (z,w)

$$
\begin{aligned}
A &:= -D_u f(z_0,u_0)^{-1} D_z f(z_0,u_0) \\
(z,u) = u(z,w) &:= (z, w + A(z-z_0))
\end{aligned}
$$

$$
f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0
$$

We want to compute

$$
N = w_0 - [D_w g(Z,W)]_I^{-1} g(Z,w_0).
$$

Compute:

$$
\begin{aligned}
D_w g(Z,W) &\subset D_u f(Z,U) \\
g(Z,w_0) &\subset g(z_0,w_0) + D_z g(Z,w_0)\cdot(Z-z_0)
\end{aligned}
$$

where

$$
\begin{aligned}
D_z g(Z,w_0) &\subset D_z f(Z,u_0) + D_u f(Z,u_0)A \\
D_z g(Z,w_0) &\subset D_z f(Z,u_0) - \left(D_u f(Z,u_0) D_u f(z_0,u_0)^{-1}\right) D_z f(z_0,u_0).
\end{aligned}
$$

Playing with coordinate systems – flatten the curve

Consider the problem $f(z,u) = 0$ and fix $z_0 \in Z$.
New coordinates (z, w)

$$A := -D_u f(z_0, u_0)^{-1} D_z f(z_0, u_0)$$
$$(z, u) = u(z, w) := (z, w + A(z - z_0))$$

$$f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0$$

We want to compute

$$N = w_0 - [D_w g(Z, W)]_I^{-1} g(Z, w_0).$$

Compute:

$$\begin{aligned} D_w g(Z, W) &\subset D_u f(Z, U) \\ g(Z, w_0) &\subset g(z_0, w_0) + D_z g(Z, w_0) \cdot (Z - z_0) \end{aligned}$$

where

$$D_z g(Z, w_0) \subset D_z f(Z, u_0) + D_u f(Z, u_0) A$$
$$D_z g(Z, w_0) \subset D_z f(Z, u_0) - \left(D_u f(Z, u_0) D_u f(z_0, u_0)^{-1} \right) D_z f(z_0, u_0).$$

Consider the problem $f(z,u)=0$ and fix $z_0 \in Z$.
New coordinates (z,w)

$$A := -D_u f(z_0,u_0)^{-1} D_z f(z_0,u_0)$$
$$(z,u) = u(z,w) := (z, w + A(z-z_0))$$

$$f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0$$

We want to compute

$$N = w_0 - [D_w g(Z,W)]_I^{-1} g(Z,w_0).$$

Compute:

$$\begin{aligned} D_w g(Z,W) &\subset D_u f(Z,U) \\ g(Z,w_0) &\subset g(z_0,w_0) + D_z g(Z,w_0)\cdot(Z-z_0) \end{aligned}$$

where

$$D_z g(Z,w_0) \subset D_z f(Z,u_0) + D_u f(Z,u_0)A$$

$$D_z g(Z,w_0) \subset D_z f(Z,u_0) - \left(D_u f(Z,u_0) D_u f(z_0,u_0)^{-1}\right) D_z f(z_0,u_0).$$

Playing with coordinate systems – flatten the curve

Consider the problem $f(z,u) = 0$ and fix $z_0 \in Z$.
New coordinates (z, w)

$$A := -D_u f(z_0, u_0)^{-1} D_z f(z_0, u_0)$$
$$(z, u) = u(z, w) := (z, w + A(z - z_0))$$

$$f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0$$

We want to compute

$$N = w_0 - [D_w g(Z, W)]_I^{-1} g(Z, w_0).$$

Compute:

$$D_w g(Z, W) \subset D_u f(Z, U)$$
$$g(Z, w_0) \subset g(z_0, w_0) + D_z g(Z, w_0) \cdot (Z - z_0)$$

where

$$D_z g(Z, w_0) \subset D_z f(Z, u_0) + D_u f(Z, u_0) A$$
$$D_z g(Z, w_0) \subset D_z f(Z, u_0) - \left(D_u f(Z, u_0) D_u f(z_0, u_0)^{-1} \right) D_z f(z_0, u_0).$$

Playing with coordinate systems – flatten the curve

Consider the problem $f(z,u) = 0$ and fix $z_0 \in Z$.
New coordinates (z, w)

$$A := -D_u f(z_0, u_0)^{-1} D_z f(z_0, u_0)$$
$$(z, u) = u(z, w) := (z, w + A(z - z_0))$$

$$f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0$$

We want to compute

$$N = w_0 - [D_w g(Z, W)]_I^{-1} g(Z, w_0).$$

Compute:

$$\begin{aligned} D_w g(Z, W) &\subset D_u f(Z, U) \\ g(Z, w_0) &\subset g(z_0, w_0) + D_z g(Z, w_0) \cdot (Z - z_0) \end{aligned}$$

where

$$\begin{aligned} D_z g(Z, w_0) &\subset D_z f(Z, u_0) + D_u f(Z, u_0) A \\ D_z g(Z, w_0) &\subset D_z f(Z, u_0) - \left(D_u f(Z, u_0) D_u f(z_0, u_0)^{-1} \right) D_z f(z_0, u_0). \end{aligned}$$

Playing with coordinate systems – flatten the curve

Consider the problem $f(z,u)=0$ and fix $z_0 \in Z$.
New coordinates (z,w)

$$A := -D_u f(z_0,u_0)^{-1} D_z f(z_0,u_0)$$
$$(z,u) = u(z,w) := (z, w + A(z-z_0))$$

$$f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0$$

We want to compute

$$N = w_0 - [D_w g(Z,W)]_I^{-1} g(Z,w_0).$$

Compute:

$$\begin{aligned} D_w g(Z,W) &\subset D_u f(Z,U) \\ g(Z,w_0) &\subset g(z_0,w_0) + D_z g(Z,w_0)\cdot(Z-z_0) \end{aligned}$$

where

$$\begin{aligned} D_z g(Z,w_0) &\subset D_z f(Z,u_0) + D_u f(Z,u_0) A \\ D_z g(Z,w_0) &\subset D_z f(Z,u_0) - \left(D_u f(Z,u_0) D_u f(z_0,u_0)^{-1} \right) D_z f(z_0,u_0). \end{aligned}$$

Playing with coordinate systems – flatten the curve

Consider the problem $f(z,u)=0$ and fix $z_0 \in Z$.
New coordinates (z,w)

$$A := -D_u f(z_0,u_0)^{-1} D_z f(z_0,u_0)$$
$$(z,u) = u(z,w) := (z, w + A(z-z_0))$$

$$f = 0 \qquad \leftrightarrow \qquad g := f \circ u = 0$$

We want to compute

$$N = w_0 - [D_w g(Z,W)]_I^{-1} g(Z,w_0).$$

Compute:

$$\begin{aligned} D_w g(Z,W) &\subset D_u f(Z,U) \\ g(Z,w_0) &\subset g(z_0,w_0) + D_z g(Z,w_0)\cdot(Z-z_0) \end{aligned}$$

where

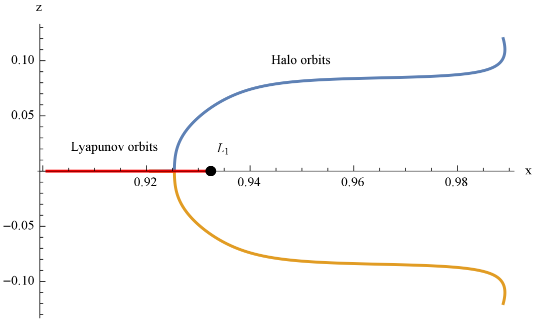
$$\begin{aligned} D_z g(Z,w_0) &\subset D_z f(Z,u_0) + D_u f(Z,u_0)A \\ D_z g(Z,w_0) &\subset D_z f(Z,u_0) - \left(D_u f(Z,u_0) D_u f(z_0,u_0)^{-1}\right) D_z f(z_0,u_0). \end{aligned}$$

Theorem (Partial result about Halo orbits)

Fix $\mu = 0.0009537$ – corresponding to Sun-Jupiter system.
There is a smooth branch of Halo orbits

$$Z \ni z \to (x(z), 0, z, 0, \dot{y}(z), 0) \in \Pi$$

$$Z := [-0.083664781253492707, 0.083664781253492707]$$



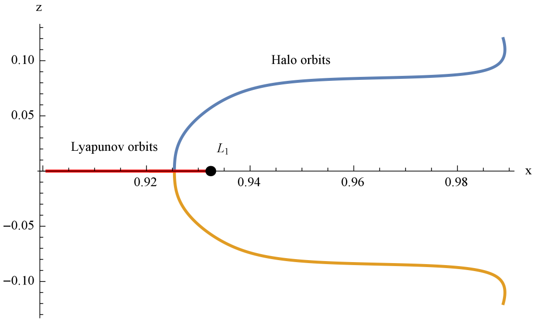
Remark: there is a bifurcation for $z = 0$. Higher order derivatives are used to get the result for $z \approx 0$ – details later.

Theorem (Partial result about Halo orbits)

Fix $\mu = 0.0009537$ *– corresponding to Sun-Jupiter system. There is a smooth branch of Halo orbits*

$$Z \ni z \to (x(z), 0, z, 0, \dot{y}(z), 0) \in \Pi$$

$$Z := [-0.083664781253492707, 0.083664781253492707]$$

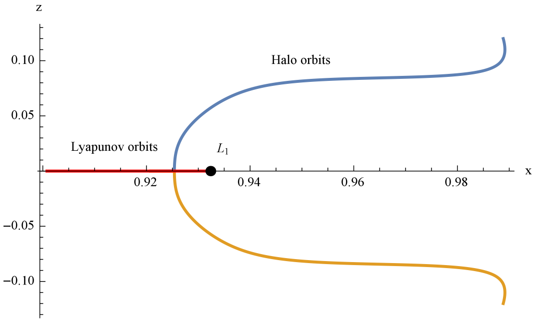


Remark: there is a bifurcation for $z = 0$. Higher order derivatives are used to get the result for $z \approx 0$ – details later.

Theorem (Partial result about Halo orbits)

Fix $\mu = 0.0009537$ *– corresponding to Sun-Jupiter system.*
There is a smooth branch of Halo orbits

$$Z \ni z \to (x(z), 0, z, 0, \dot{y}(z), 0) \in \Pi$$
$$Z := [-0.083664781253492707, 0.083664781253492707]$$



Remark: there is a bifurcation for $z = 0$. Higher order derivatives are used to get the result for $z \approx 0$ – details later.

Boundary value problems for ODEs of any type

Second order equation:

$$x'' = f(x, x')$$

BVPs can be transformed to $F(u) = 0$

Dirichlet BVP:

$$x(0) = A,\ x(T) = B \qquad \rightsquigarrow \qquad F(x') = \pi_x\left(\varphi(T, (A, x'))\right) - B$$

Neumann BVP:

$$x'(0) = A,\ x'(T) = B \qquad \rightsquigarrow \qquad F(x) = \pi_{x'}\left(\varphi(T, (x, A))\right) - B$$

Solve by interval Newton (Krawczyk) method

Boundary value problems for ODEs of any type

Second order equation:

$$x'' = f(x, x')$$

BVPs can be transformed to $F(u) = 0$

Dirichlet BVP:

$$x(0) = A,\ x(T) = B \qquad \rightsquigarrow \qquad F(x') = \pi_x\left(\varphi(T, (A, x'))\right) - B$$

Neumann BVP:

$$x'(0) = A,\ x'(T) = B \qquad \rightsquigarrow \qquad F(x) = \pi_{x'}\left(\varphi(T, (x, A))\right) - B$$

Solve by interval Newton (Krawczyk) method

Boundary value problems for ODEs of any type

Second order equation:

$$x'' = f(x, x')$$

BVPs can be transformed to $F(u) = 0$

Dirichlet BVP:

$$x(0) = A,\ x(T) = B \qquad \rightsquigarrow \qquad F(x') = \pi_x \left(\varphi(T, (A, x'))\right) - B$$

Neumann BVP:

$$x'(0) = A,\ x'(T) = B \qquad \rightsquigarrow \qquad F(x) = \pi_{x'} \left(\varphi(T, (x, A))\right) - B$$

Solve by interval Newton (Krawczyk) method

Boundary value problems for ODEs of any type

Second order equation:

$$x'' = f(x, x')$$

BVPs can be transformed to $F(u) = 0$

Dirichlet BVP:

$$x(0) = A,\ x(T) = B \qquad \rightsquigarrow \qquad F(x') = \pi_x\left(\varphi(T, (A, x'))\right) - B$$

Neumann BVP:

$$x'(0) = A,\ x'(T) = B \qquad \rightsquigarrow \qquad F(x) = \pi_{x'}\left(\varphi(T, (x, A))\right) - B$$

Solve by interval Newton (Krawczyk) method

Boundary value problems for ODEs of any type

Second order equation:

$$x'' = f(x, x')$$

BVPs can be transformed to $F(u) = 0$

Dirichlet BVP:

$$x(0) = A,\ x(T) = B \qquad \rightsquigarrow \qquad F(x') = \pi_x\left(\varphi(T, (A, x'))\right) - B$$

Neumann BVP:

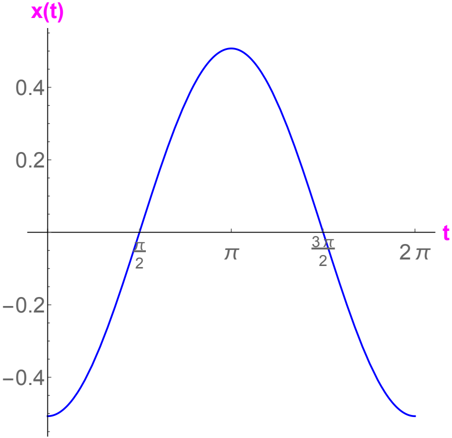
$$x'(0) = A,\ x'(T) = B \qquad \rightsquigarrow \qquad F(x) = \pi_{x'}\left(\varphi(T, (x, A))\right) - B$$

Solve by interval Newton (Krawczyk) method

Example (Taken from Nakao, J. Math. Anal. App. 1992)

$$x'' = -0.1x - 0.1x^3 - 0.4464\cos t$$

Find solution to $x'(0) = x'(2\pi) = 0$.



Method: solve using interval Newton method

$$F(r_1) = \varphi_{\dot{x}}(2\pi, (-0.5072 + r_1, 0)) = 0$$

```cpp
/** Example of solving BVP: x'(0)=x'(2pi)=0 **/
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
int main(){
  IMap f("par:a;time:t;var:x,dx;fun:dx,-x*(1+x^2)/10 + a*cos(t);");
  f.setParameter("a",interval(4464)/interval(10000));
  IOdeSolver solver(f,20);  // ODE integrator
  ITimeMap tm(solver);      // class for long time integration

  IVector u0({-0.5072,0.});
  IVector r({interval(-1e-5,1e-5),0.});
  C0HOTripletonSet s0(u0);
  C1HORect2Set s(u0+r);
  // integrate until T=2*pi
  IVector y = tm(2.*interval::pi(),s0);
  tm(2.*interval::pi(),s);
  // solve equation F(r1) := proj_{x'}(phi(2pi,u0+(r1,0))) = 0
  interval N = - y[1]/((IMatrix)s)(2,1);
  cout << "(N,r1)=(" << N << "," << r[0] << ")"<< endl;
  cout << "subset(N,r)? = " << boolalpha << subset(N,r[0]) << endl;
  return 0;
}
/* Output:
(N,r1)=([-3.84493e-05, -2.09823e-05],[-0.1, 0.1])
subset(N,r)? = true
*/
```

Hyperbolic dynamics

- hyperbolic horseshoe in the Rössler system
- hyperbolic chaotic attractor in the Kuznetsov system

Definition

f is uniformly hyperbolic on M iff

- $TM = E^u \oplus E^s$ and E^u, E^s are Tf-invariant subbundles

$$Df(x)(E^u_x) = E^u_{f(x)}, \qquad Df(x)(E^s_x) = E^s_{f(x)}, \quad \text{for } x \in M$$

- There are constants $c > 0$ and $0 < \lambda < 1$ such that

$$\|Df^n(x)v\| < c\lambda^n\|v\|, \quad \text{for } v \in E^s_x,\ x \in M$$
$$\|Df^{-n}(x)v\| < c\lambda^n\|v\|, \quad \text{for } v \in E^u_x,\ x \in M$$

Definition

f is uniformly hyperbolic on M iff

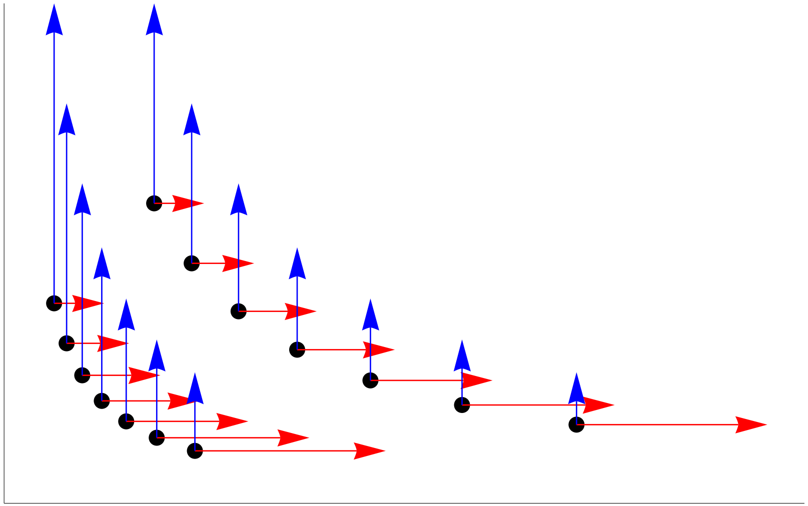
- $TM = E^u \oplus E^s$ and E^u, E^s are Tf-invariant subbundles

$$Df(x)(E^u_x) = E^u_{f(x)}, \qquad Df(x)(E^s_x) = E^s_{f(x)}, \quad \text{for } x \in M$$

- There are constants $c > 0$ and $0 < \lambda < 1$ such that

$$\|Df^n(x)v\| < c\lambda^n\|v\|, \quad \text{for } v \in E^s_x,\ x \in M$$
$$\|Df^{-n}(x)v\| < c\lambda^n\|v\|, \quad \text{for } v \in E^u_x,\ x \in M$$

Invariant subbundles



Examples of hyperbolic invariant sets

- hyperbolic fixed point or periodic orbit.
- Finite union of periodic points.
- Arnold's cat map - diffeomorpshism of the torus $\mathcal{S}^1 \times \mathcal{S}^1$

$$(x, y) \to (2x + y, x + y)$$

- hyperbolic fixed point or periodic orbit.
- Finite union of periodic points.

- Arnold's cat map - diffeomorpshism of the torus $\mathcal{S}^1 \times \mathcal{S}^1$

$$(x, y) \to (2x + y, x + y)$$

Examples of hyperbolic invariant sets

- hyperbolic fixed point or periodic orbit.
- Finite union of periodic points.

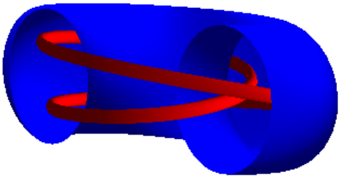
- Arnold's cat map - diffeomorpshism of the torus $\mathcal{S}^1 \times \mathcal{S}^1$

$$(x, y) \to (2x + y, x + y)$$

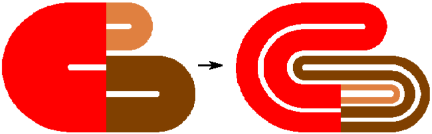
Examples of hyperbolic attractors

Smale map - defined on a solid torus $\mathbb{T} = D^2 \times \mathcal{S}^1$

$$s \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} = \begin{bmatrix} 0.1x + 0.5\cos\alpha \\ 0.1y + 0.5\sin\alpha \\ 2\alpha \end{bmatrix}$$



Plykin attractor originally defined on the sphere $\mathcal{S}^2$.



Some properties of locally maximal hyperbolic sets

Definition

Λ is locally maximal hyperbolic set for f iff Λ is hyperbolic and there is an open ngbh U of Λ such that $\Lambda = \bigcap_{n \in \mathbb{Z}} f^n(U)$.

- structural stability: persist under any $\mathcal{C}^1$ perturbation of f; no bifurcation occur when the map is $\mathcal{C}^1$ perturbed
- entropy and number of periodic points persist under perturbation
- periodic points are dense in the set of non-wandering points (Anosov Closing Lemma)

Some properties of locally maximal hyperbolic sets

Definition

Λ is locally maximal hyperbolic set for f iff Λ is hyperbolic and there is an open ngbh U of Λ such that $\Lambda = \bigcap_{n\in\mathbb{Z}} f^n(U)$.

- structural stability: persist under any $\mathcal{C}^1$ perturbation of f; no bifurcation occur when the map is $\mathcal{C}^1$ perturbed
- entropy and number of periodic points persist under perturbation
- periodic points are dense in the set of non-wandering points (Anosov Closing Lemma)

Some properties of locally maximal hyperbolic sets

Definition

Λ is locally maximal hyperbolic set for f iff Λ is hyperbolic and there is an open ngbh U of Λ such that $\Lambda = \bigcap_{n\in\mathbb{Z}} f^n(U)$.

- structural stability: persist under any $\mathcal{C}^1$ perturbation of f; no bifurcation occur when the map is $\mathcal{C}^1$ perturbed
- entropy and number of periodic points persist under perturbation
- periodic points are dense in the set of non-wandering points (Anosov Closing Lemma)

Some theoretical tools for proving hyperbolicity

Z. Galias, P. Zgliczyński, Abundance of homoclinic and heteroclinic orbitsand rigorous bounds for the topological entropy for the Henon map, Nonlinearity, 14 (2001), 909-932

S.L. Hruska, A Numerical Method for Constructing the Hyperbolic Structure of Complex Hénon Mappings, Found. Comp. Math. 6, No. 4, (2006), 427–455.

S.L. Hruska, Rigorous numerical models for the dynamics of complex Hénon mappings on their chain recurrent sets, Discrete Cont. Dyn. Sys., 15 (2) (2006), 529–558.

Z. Arai, *On Hyperbolic Plateaus of the Hénon map*, Experimental Mathematics, 16:2 (2007), 181–188.

H. Kokubu, DW, P. Zgliczyński, Rigorous verification of cocoon bifurcations in the Michelson system, Nonlinearity 20 (2007) 2147-2174

M. Mazur, J. Tabor, P. Kościelniak, *Semi-hyperbolicity and hyperbolicity*, Disc. Cont. Dyn. Sys. 20, No. 4 (2008), 1029–1038.

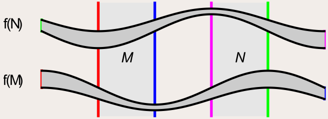
P. Zgliczyński, Covering relations, cone conditions and stable manifold theorem, Journal of Differential Equations 246, (2009) 1774–1819.

DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, SIADS 2010, Vol. 9, 1263–1283.

M. Mazur, J. Tabor, *Computational hyperbolicity*, Disc. Cont. Dyn. Sys. A, 29:3 (2011) 1175–1189.

Theorem

- *Let* $Q = DiagonalMatrix\{\lambda, \mu\}$ *with* $\mu < 0 < \lambda$
- N, M *disjoint rectangles aligned to the axes*
- $f : N \cup M \to \mathbb{R}^2$ *is smooth*



- $N \xrightarrow{f} N \xrightarrow{f} M \xrightarrow{f} M \xrightarrow{f} N.$
- $Df(x)^T QDf(x) - Q$ *is positive definite for* $x \in N \cup M$

Then

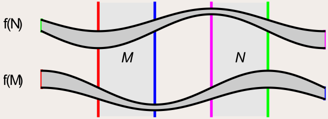
- $\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ *is uniformly hyperbolic and the dynamics of* f *on* $\mathcal{H}$ *is conjugated to the full shift on two symbols*
- *every binifinite sequence of symbols of the form*

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ *and* $p_B \in B$

Theorem

- *Let* $Q = \mathit{DiagonalMatrix}\{\lambda, \mu\}$ *with* $\mu < 0 < \lambda$
- N, M *disjoint rectangles aligned to the axes*
- $f : N \cup M \to \mathbb{R}^2$ *is smooth*



- $N \xrightarrow{f} N \xrightarrow{f} M \xrightarrow{f} M \xrightarrow{f} N.$
- $Df(x)^T Q Df(x) - Q$ *is positive definite for* $x \in N \cup M$

Then

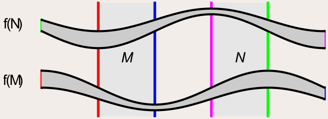
- $\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ *is uniformly hyperbolic and the dynamics of* f *on* $\mathcal{H}$ *is conjugated to the full shift on two symbols*
- *every binifinite sequence of symbols of the form*

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ *and* $p_B \in B$

Theorem

- *Let $Q = DiagonalMatrix\{\lambda, \mu\}$ with $\mu < 0 < \lambda$*
- *N, M disjoint rectangles aligned to the axes*
- *$f : N \cup M \to \mathbb{R}^2$ is smooth*



- $N \xrightarrow{f} N \xrightarrow{f} M \xrightarrow{f} M \xrightarrow{f} N.$
- *$Df(x)^T QDf(x) - Q$ is positive definite for $x \in N \cup M$*

Then

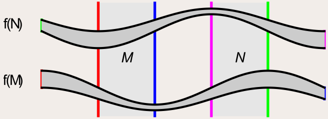
- *$\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ is uniformly hyperbolic and the dynamics of f on $\mathcal{H}$ is conjugated to the full shift on two symbols*
- *every binifinite sequence of symbols of the form*

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ and $p_B \in B$

Theorem

- *Let* $Q = DiagonalMatrix\{\lambda, \mu\}$ *with* $\mu < 0 < \lambda$
- N, M *disjoint rectangles aligned to the axes*
- $f : N \cup M \to \mathbb{R}^2$ *is smooth*



- $N \overset{f}{\Longrightarrow} N \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} N.$
- $Df(x)^T Q Df(x) - Q$ *is positive definite for* $x \in N \cup M$

Then

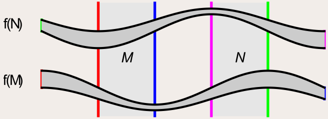
- $\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ *is uniformly hyperbolic and the dynamics of* f *on* $\mathcal{H}$ *is conjugated to the full shift on two symbols*
- *every binifinite sequence of symbols of the form*

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ *and* $p_B \in B$

Theorem

- Let $Q = DiagonalMatrix\{\lambda, \mu\}$ with $\mu < 0 < \lambda$
- N, M disjoint rectangles aligned to the axes
- $f : N \cup M \to \mathbb{R}^2$ is smooth
- $N \overset{f}{\Longrightarrow} N \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} N.$



- $Df(x)^T Q Df(x) - Q$ is positive definite for $x \in N \cup M$

Then

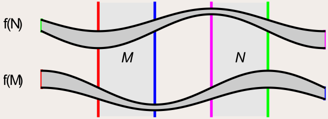
- $\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ is uniformly hyperbolic and the dynamics of f on $\mathcal{H}$ is conjugated to the full shift on two symbols
- every binifinite sequence of symbols of the form

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ and $p_B \in B$

Theorem

- *Let* $Q = \textit{DiagonalMatrix}\{\lambda, \mu\}$ *with* $\mu < 0 < \lambda$
- N, M *disjoint rectangles aligned to the axes*
- $f : N \cup M \to \mathbb{R}^2$ *is smooth*
- $N \overset{f}{\Longrightarrow} N \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} N.$



- $Df(x)^T Q Df(x) - Q$ *is positive definite for* $x \in N \cup M$

Then

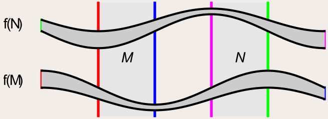
- $\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ *is uniformly hyperbolic and the dynamics of* f *on* $\mathcal{H}$ *is conjugated to the full shift on two symbols*
- *every binifinite sequence of symbols of the form*

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ *and* $p_B \in B$

Theorem

- Let $Q = DiagonalMatrix\{\lambda, \mu\}$ with $\mu < 0 < \lambda$
- N, M disjoint rectangles aligned to the axes
- $f : N \cup M \to \mathbb{R}^2$ is smooth
- $N \overset{f}{\Longrightarrow} N \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} M \overset{f}{\Longrightarrow} N$.



- $Df(x)^T QDf(x) - Q$ is positive definite for $x \in N \cup M$

Then

- $\mathcal{H} = \mathrm{Inv}(f, N \cup M)$ is uniformly hyperbolic and the dynamics of f on $\mathcal{H}$ is conjugated to the full shift on two symbols
- every binifinite sequence of symbols of the form

$$(\ldots, A, A, \{N, M\}^k, B, B, \ldots), \qquad A, B \in \{N, M\}$$

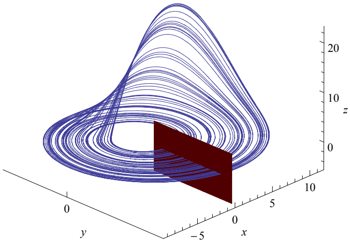
is realized by a heteroclinic (or homoclinic) orbit connecting unique fixed point $p_A \in A$ and $p_B \in B$

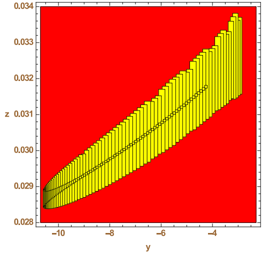
Example (Hyperbolic chaos in the Rössler system)

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

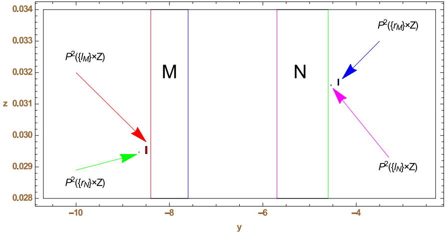
We already proved:

- there is a connected and compact attractor $\mathcal{A}$





- there is an invariant subset $\mathcal{H} \subset \mathcal{A}$ on which the system is chaotic

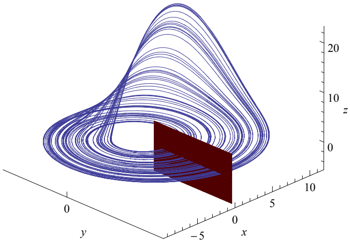


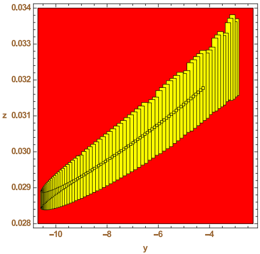
Example (Hyperbolic chaos in the Rössler system)

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

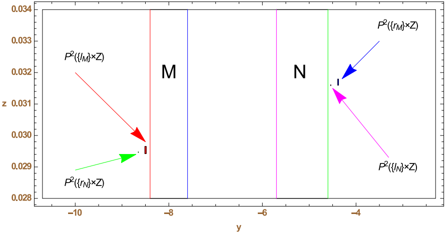
We already proved:

- there is a connected and compact attractor $\mathcal{A}$





- there is an invariant subset $\mathcal{H} \subset \mathcal{A}$ on which the system is chaotic



Goal 1:

Dynamics on $\mathcal{H}$ is uniformly hyperbolic with one positive and one negative Lyapunov exponents. In particular all periodic orbits in $\mathcal{H}$ are hyperbolic

Goal 2:

There are two hyperbolic periodic orbits $p_N, p_M \in \mathcal{H}$ with low periods

Goal 3:

There is countable infinity of heteroclinic connection between these orbits in both directions

Goal 4:

There is countable infinity of homoclinic orbits to both periodic orbits p_N and p_M.

Goal 1:
Dynamics on $\mathcal{H}$ is uniformly hyperbolic with one positive and one negative Lyapunov exponents. In particular all periodic orbits in $\mathcal{H}$ are hyperbolic

Goal 2:
There are two hyperbolic periodic orbits $p_N, p_M \in \mathcal{H}$ with low periods

Goal 3:
There is countable infinity of heteroclinic connection between these orbits in both directions

Goal 4:
There is countable infinity of homoclinic orbits to both periodic orbits p_N and p_M.

Goal 1:
Dynamics on $\mathcal{H}$ is uniformly hyperbolic with one positive and one negative Lyapunov exponents. In particular all periodic orbits in $\mathcal{H}$ are hyperbolic

Goal 2:
There are two hyperbolic periodic orbits $p_N, p_M \in \mathcal{H}$ with low periods

Goal 3:
There is countable infinity of heteroclinic connection between these orbits in both directions

Goal 4:
There is countable infinity of homoclinic orbits to both periodic orbits p_N and p_M.

Goal 1:
Dynamics on $\mathcal{H}$ is uniformly hyperbolic with one positive and one negative Lyapunov exponents. In particular all periodic orbits in $\mathcal{H}$ are hyperbolic

Goal 2:
There are two hyperbolic periodic orbits $p_N, p_M \in \mathcal{H}$ with low periods

Goal 3:
There is countable infinity of heteroclinic connection between these orbits in both directions

Goal 4:
There is countable infinity of homoclinic orbits to both periodic orbits p_N and p_M.

Methodology: check the cone conditions

$$DP^2(x)^T \cdot Q \cdot DP^2(x) - Q > 0$$

for $x \in N \cup M$.

Abstract theorem guarantees

- every biinfinite sequence of symbols

$$\{N, M\}^{\mathbb{Z}}$$

  is realized by a unique orbit. In particular there is an unique fixed point for P^2 in N and an unique fixed point for P^2 in M
- every “heteroclinic” chain of symbols is realized by a heteroclinic orbit
- dynamics is uniformly hyperbolic

Methodology: check the cone conditions

$$DP^2(x)^T \cdot Q \cdot DP^2(x) - Q > 0$$

for $x \in N \cup M$.

Abstract theorem guarantees

- every biinfinite sequence of symbols

$$\{N, M\}^{\mathbb{Z}}$$

  is realized by a unique orbit. In particular there is an unique fixed point for P^2 in N and an unique fixed point for P^2 in M
- every "heteroclinic" chain of symbols is realized by a heteroclinic orbit
- dynamics is uniformly hyperbolic

Methodology: check the cone conditions

$$DP^2(x)^T \cdot Q \cdot DP^2(x) - Q > 0$$

for $x \in N \cup M$.

Abstract theorem guarantees

- every biinfinite sequence of symbols

$$\{N, M\}^{\mathbb{Z}}$$

  is realized by a unique orbit. In particular there is an unique fixed point for P^2 in N and an unique fixed point for P^2 in M
- every “heteroclinic” chain of symbols is realized by a heteroclinic orbit
- dynamics is uniformly hyperbolic

Methodology: check the cone conditions

$$DP^2(x)^T \cdot Q \cdot DP^2(x) - Q > 0$$

for $x \in N \cup M$.

Abstract theorem guarantees

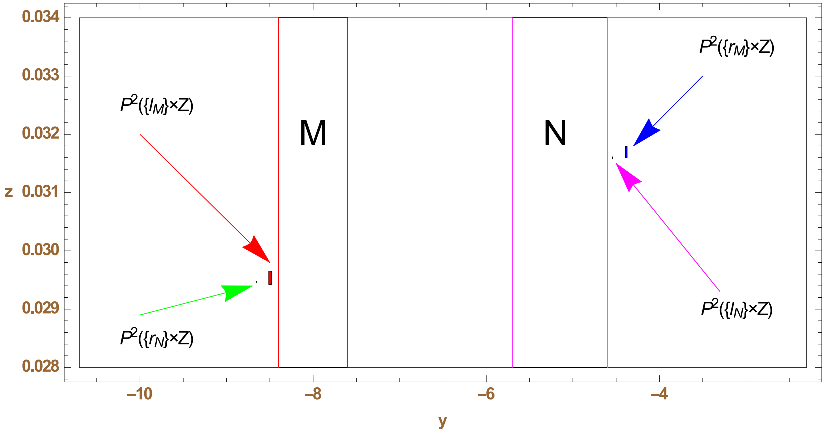
- every biinfinite sequence of symbols

  $$\{N, M\}^{\mathbb{Z}}$$

  is realized by a unique orbit. In particular there is an unique fixed point for P^2 in N and an unique fixed point for P^2 in M
- every "heteroclinic" chain of symbols is realized by a heteroclinic orbit
- dynamics is uniformly hyperbolic

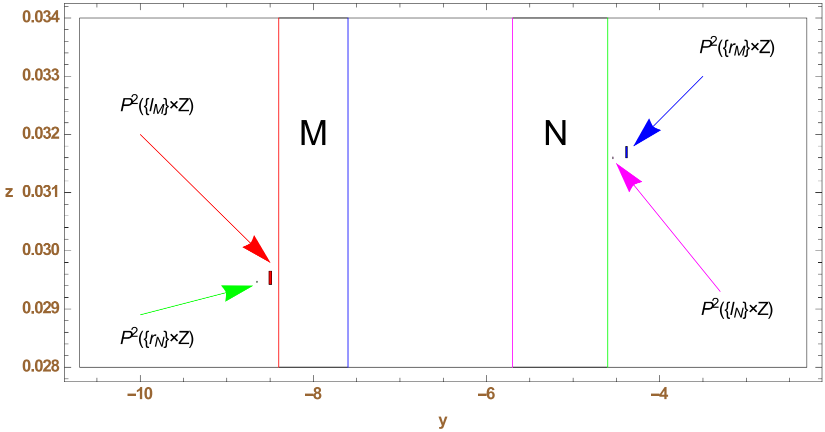
Data:

$$M = [l_M, r_M] \times Z = [-8.4, -7.6] \times [0.028, 0.034]$$
$$N = [l_N, r_N] \times Z = [-5.7, -4.6] \times [0.028, 0.034]$$
$$Q = \texttt{DiagonalMatrix}\{1, -100\}$$



Data:

$$M = [l_M, r_M] \times Z = [-8.4, -7.6] \times [0.028, 0.034]$$
$$N = [l_N, r_N] \times Z = [-5.7, -4.6] \times [0.028, 0.034]$$
$$Q = \texttt{DiagonalMatrix}\{1, -100\}$$



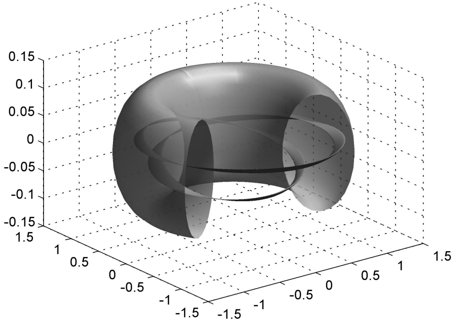
```cpp
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
bool checkCC(IPoincareMap& pm, double y1, double y2, int N) {
  bool res = true;
  interval p = (interval(y2) - interval(y1)) / N;
  IMatrix Dphi(3,3);
  IMatrix Q({{0.,0.,0.},{0.,1,0.},{0.,0.,-100}});
  interval returnTime;
  for (int i = 0; i < N and res; ++i) {
    C1Rect2Set s({0.,y1+interval(i,i+1)*p,interval(0.028,0.034)});
    IVector y = pm(s, Dphi, returnTime, 2);
    IMatrix DP = pm.computeDP(y,Dphi);
    DP = Transpose(DP)*Q*DP - Q;
    res = res and DP(2,2)>0 and (DP(2,2)*DP(3,3)-sqr(DP(2,3)))>0;
  }
  return res;
}
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);
  const double lM=-8.4, rM=-7.6, lN=-5.7, rN=-4.6;
  cout << "Cone condition on M: " << checkCC(pm,lM,rM,80) << endl;
  cout << "Cone condition on N: " << checkCC(pm,lN,rN,20) << endl;
}
```

Example (Kuznetsov system)

$$\begin{cases} \dot{x} &= \omega_0 u, \\ \dot{u} &= -\omega_0 x + \left(A\cos(2\pi t/T) - x^2\right) u + (\varepsilon/\omega_0) y \cos(\omega_0 t), \\ \dot{y} &= 2\omega_0 v, \\ \dot{v} &= -2\omega_0 y + \left(-A\cos(2\pi t/T) - y^2\right) v + (\varepsilon/2\omega_0) x^2. \end{cases}$$

$\omega_0 = 2\pi$, $A = 5$, $T = 6$, $\varepsilon = 0.5$

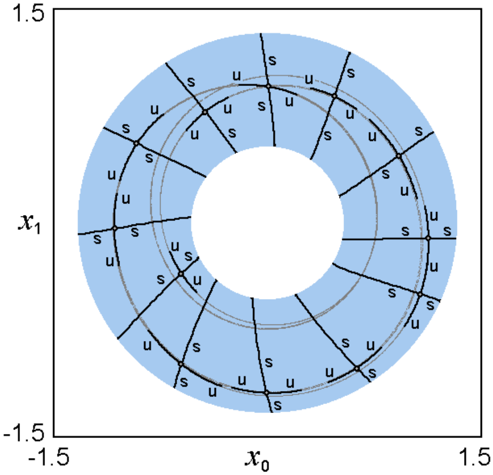
Click here to start animation



S.P. Kuznetsov, Example of a Physical System with a Hyperbolic Attractor of the Smale-Williams Type, Phys. Rev. Lett., 95, 2005, 144101.

Conjecture: the system has hyperbolic attractor for

$$\omega_0 = 2\pi, \quad A = 5, \quad T = 6, \quad \varepsilon = 0.5$$



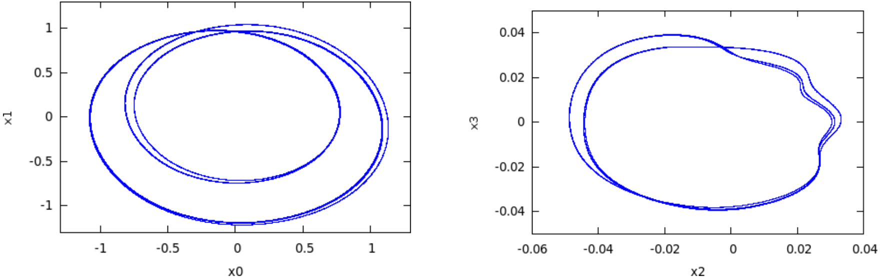
S.P. Kuznetsov and I.R. Sataev, Hyperbolic attractor in a system of coupled non-autonomous van der Pol oscillators: Numerical test for expanding and contracting cones, Phys. Lett. A 365, 97–104, (2007).

Theorem

Poincaré map: $P(x) = \varphi(T, x)$, $T = 6$ *(period of vector field).*

There is a compact, connected and explicitly given set $\mathcal{B}$ such that

1. $P(\mathcal{B}) \subset \mathcal{B}$,
2. P *is uniformly hyperbolic on* $\mathcal{A} = \bigcap_{i>0} P^i(\mathcal{B})$ *with one positive and three negative Lyapunov exponents.*
3. $\mathcal{A}$ *is nontrivial continuum.*

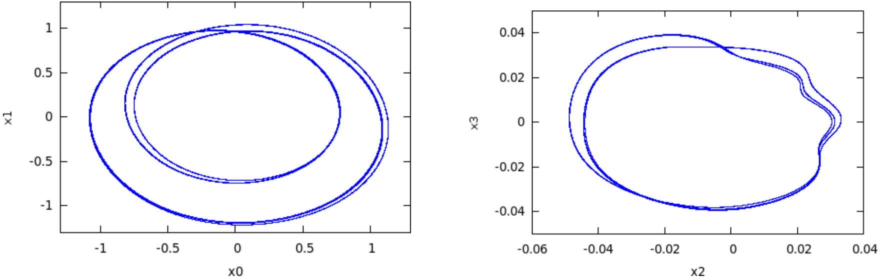


DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, SIAM J. App. Dyn. Sys. 2010, Vol. 9, 1263–1283.

Theorem

Poincaré map: $P(x) = \varphi(T, x)$, $T = 6$ (period of vector field).
There is a compact, connected and explicitly given set $\mathcal{B}$ such that

1. $P(\mathcal{B}) \subset \mathcal{B}$,
2. P *is uniformly hyperbolic on* $\mathcal{A} = \bigcap_{i>0} P^i(\mathcal{B})$ *with one positive and three negative Lyapunov exponents.*
3. $\mathcal{A}$ *is nontrivial continuum.*

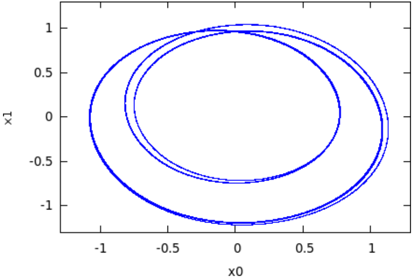


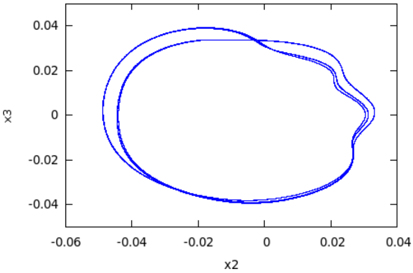
DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, SIAM J. App. Dyn. Sys. 2010, Vol. 9, 1263–1283.

Theorem

Poincaré map: $P(x) = \varphi(T, x)$, $T = 6$ (period of vector field). There is a compact, connected and explicitly given set $\mathcal{B}$ such that

1. $P(\mathcal{B}) \subset \mathcal{B}$,
2. *P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i>0} P^i(\mathcal{B})$ with one positive and three negative Lyapunov exponents.*
3. *$\mathcal{A}$ is nontrivial continuum.*



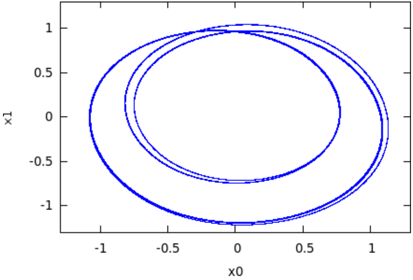


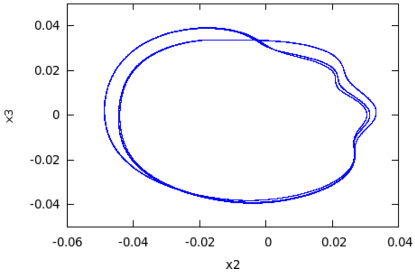
DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, SIAM J. App. Dyn. Sys. 2010, Vol. 9, 1263–1283.

Theorem

Poincaré map: $P(x) = \varphi(T, x)$, $T = 6$ (period of vector field). There is a compact, connected and explicitly given set $\mathcal{B}$ such that

1. $P(\mathcal{B}) \subset \mathcal{B}$,
2. *P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i>0} P^i(\mathcal{B})$ with one positive and three negative Lyapunov exponents.*
3. *$\mathcal{A}$ is nontrivial continuum.*



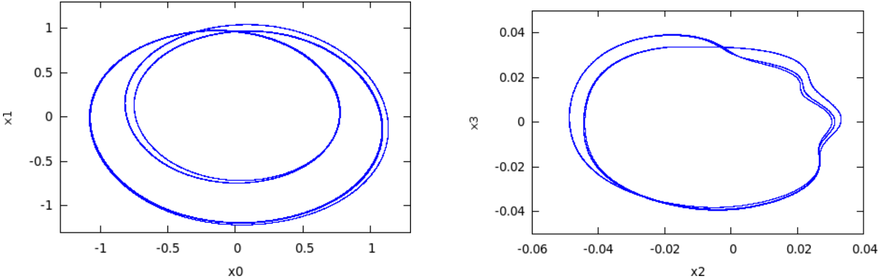


DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, SIAM J. App. Dyn. Sys. 2010, Vol. 9, 1263–1283.

Theorem

Poincaré map: $P(x) = \varphi(T, x)$, $T = 6$ (period of vector field). There is a compact, connected and explicitly given set $\mathcal{B}$ such that

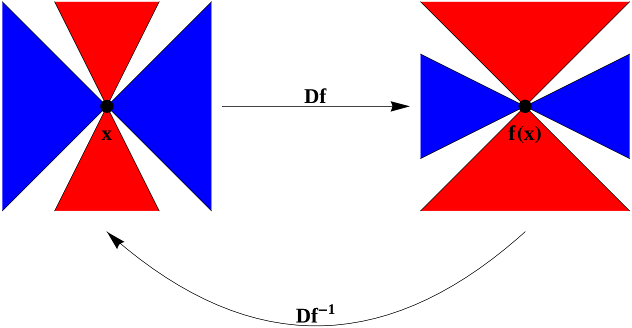
1. $P(\mathcal{B}) \subset \mathcal{B}$,
2. *P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i>0} P^i(\mathcal{B})$ with one positive and three negative Lyapunov exponents.*
3. *$\mathcal{A}$ is nontrivial continuum.*



DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, SIAM J. App. Dyn. Sys. 2010, Vol. 9, 1263–1283.

Cone criterion for hyperbolicity.

- the cones are invariant under Df and Df^{-1}, respectively
- uniform expansion and contraction in cones

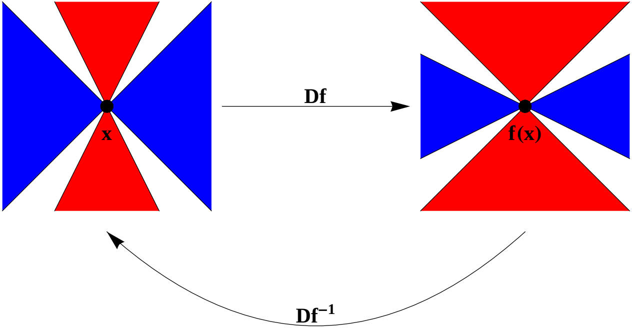


Problem:

Computing inverse map for strongly dissipative systems is at least difficult, perhaps impossible.
We need a theoretical tool for proving hyperbolicity which **does not use inverse map**.

Cone criterion for hyperbolicity.

- the cones are invariant under Df and Df^{-1}, respectively
- uniform expansion and contraction in cones

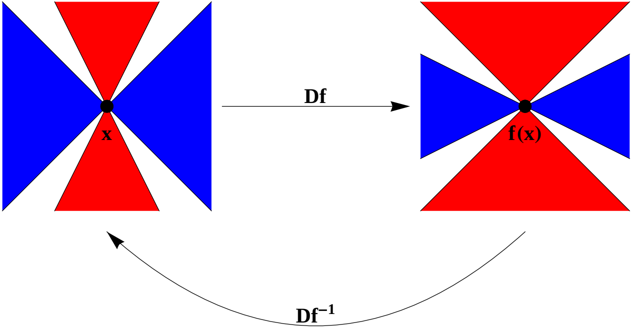


Problem:

Computing inverse map for strongly dissipative systems is at least difficult, perhaps impossible.
We need a theoretical tool for proving hyperbolicity which **does not use inverse map**.

Cone criterion for hyperbolicity.

- the cones are invariant under Df and Df^{-1}, respectively
- uniform expansion and contraction in cones

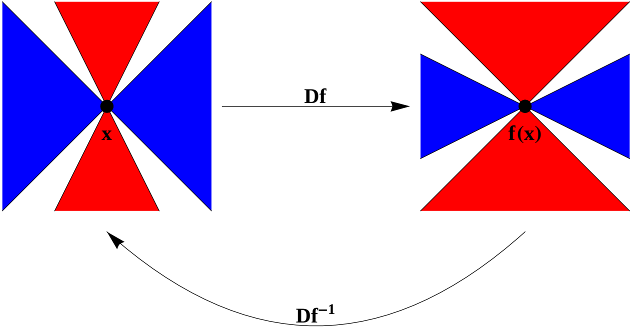


Problem:

Computing inverse map for strongly dissipative systems is at least difficult, perhaps impossible.
We need a theoretical tool for proving hyperbolicity which **does not use inverse map**.

Cone criterion for hyperbolicity.

- the cones are invariant under Df and Df^{-1}, respectively
- uniform expansion and contraction in cones



Problem:

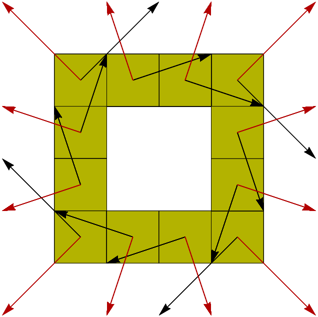
Computing inverse map for strongly dissipative systems is at least difficult, perhaps impossible.
We need a theoretical tool for proving hyperbolicity which **does not use inverse map.**

General settings

- $M = \bigcup_{i=1}^{N} M_i$, where M_i are compact sets in $\mathbb{R}^n$.
- u, s - nonnegative, such that $n = u + s$
- C_i - a linear coordinate system assigned to the set M_i
- quadratic form on $\mathbb{R}^n$

$$Q(x, y) = \|x\|^2 - \|y\|^2, \quad x \in \mathbb{R}^u,\ y \in \mathbb{R}^s.$$

$\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ is called **cubical set with cones**.



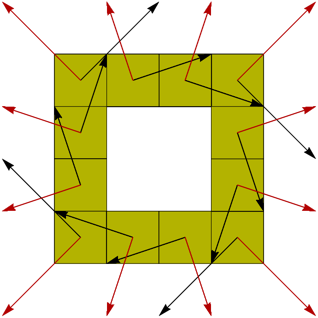
Theoretical background

General settings

- $M = \bigcup_{i=1}^{N} M_i$, where M_i are compact sets in $\mathbb{R}^n$.
- u, s - nonnegative, such that $n = u + s$
- C_i - a linear coordinate system assigned to the set M_i
- quadratic form on $\mathbb{R}^n$

$$Q(x, y) = \|x\|^2 - \|y\|^2, \quad x \in \mathbb{R}^u,\ y \in \mathbb{R}^s.$$

$\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ is called **cubical set with cones**.

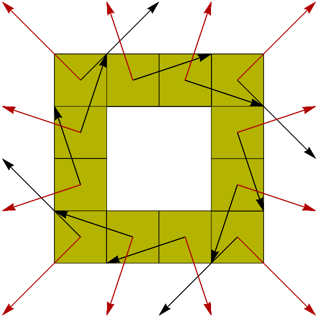


General settings

- $M = \bigcup_{i=1}^{N} M_i$, where M_i are compact sets in $\mathbb{R}^n$.
- u, s - nonnegative, such that $n = u + s$
- C_i - a linear coordinate system assigned to the set M_i
- quadratic form on $\mathbb{R}^n$

$$Q(x, y) = \|x\|^2 - \|y\|^2, \quad x \in \mathbb{R}^u,\ y \in \mathbb{R}^s.$$

$\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ is called **cubical set with cones**.

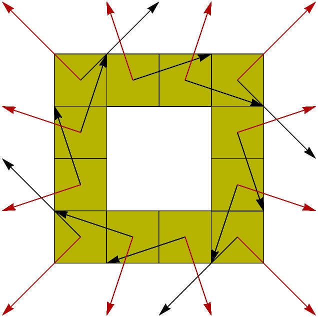


General settings

- $M = \bigcup_{i=1}^{N} M_i$, where M_i are compact sets in $\mathbb{R}^n$.
- u, s - nonnegative, such that $n = u + s$
- C_i - a linear coordinate system assigned to the set M_i
- quadratic form on $\mathbb{R}^n$

$$Q(x, y) = \|x\|^2 - \|y\|^2, \quad x \in \mathbb{R}^u,\ y \in \mathbb{R}^s.$$

$\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ is called **cubical set with cones**.



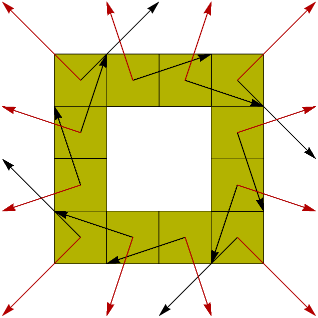
Theoretical background

General settings

- $M = \bigcup_{i=1}^{N} M_i$, where M_i are compact sets in $\mathbb{R}^n$.
- u, s - nonnegative, such that $n = u + s$
- C_i - a linear coordinate system assigned to the set M_i
- quadratic form on $\mathbb{R}^n$

$$Q(x, y) = \|x\|^2 - \|y\|^2, \quad x \in \mathbb{R}^u,\ y \in \mathbb{R}^s.$$

$\mathcal{M} = (Q, \{(M_i, C_i)\}_{i=1}^{N})$ is called **cubical set with cones**.



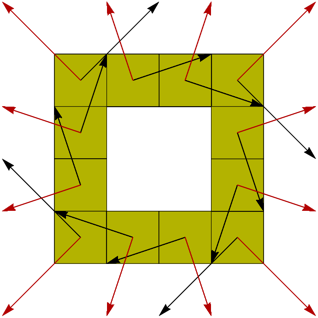
Theoretical background

General settings

- $M = \bigcup_{i=1}^{N} M_i$, where M_i are compact sets in $\mathbb{R}^n$.
- u, s - nonnegative, such that $n = u + s$
- C_i - a linear coordinate system assigned to the set M_i
- quadratic form on $\mathbb{R}^n$

$$Q(x, y) = \|x\|^2 - \|y\|^2, \quad x \in \mathbb{R}^u, \; y \in \mathbb{R}^s.$$

$\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ is called **cubical set with cones**.

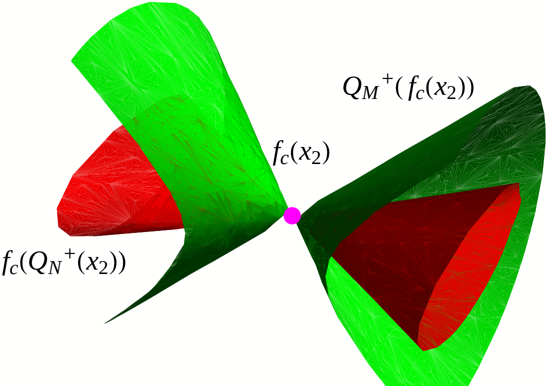


Definition

f is **strongly hyperbolic** on $\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ if for $z \in M_i$ and $j = 1, \ldots, N$ such that $f(M_i) \cap M_j \neq \emptyset$ the matrix

$$[C_j Df(z) C_i^{-1}]^T Q [C_j Df(z) C_i^{-1}] - Q$$

is positive definite.

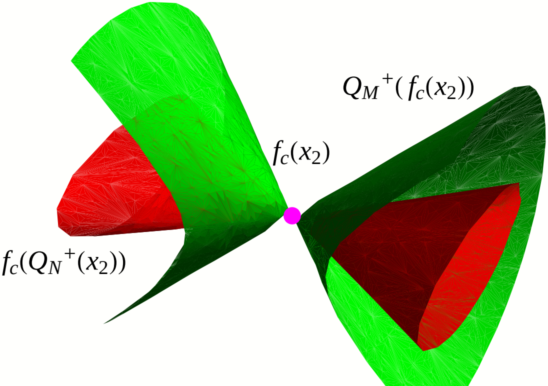


Definition

f is **strongly hyperbolic** on $\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ if for $z \in M_i$ and $j = 1, \ldots, N$ such that $f(M_i) \cap M_j \neq \emptyset$ the matrix

$$[C_j Df(z) C_i^{-1}]^T Q [C_j Df(z) C_i^{-1}] - Q$$

is positive definite.

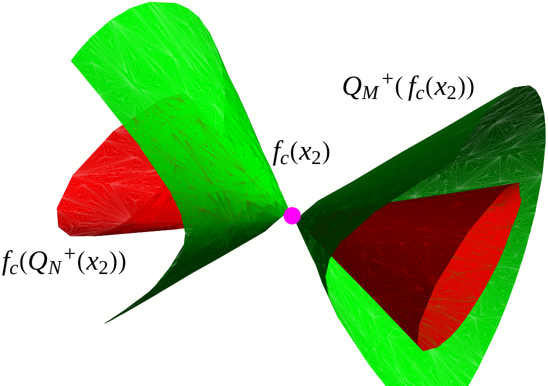


Definition

f is **strongly hyperbolic** on $\mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^{N}\right)$ if for $z \in M_i$ and $j = 1, \ldots, N$ such that $f(M_i) \cap M_j \neq \emptyset$ the matrix

$$[C_j Df(z) C_i^{-1}]^T Q [C_j Df(z) C_i^{-1}] - Q$$

is positive definite.



Main tool for hyperbolicity

Theorem

Let

$$\mathcal{H} := \mathrm{Inv}(f, M) = \left\{ x \in M : f^i(x) \in M, \text{ for } i \in \mathbb{Z} \right\}.$$

Then

$$f \text{ is strongly hyperbolic on } \mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^N \right)$$

$$\Downarrow$$

$$f \text{ is uniformly hyperbolic on } \mathcal{H}$$

Theorem

Let

$$\mathcal{H} := \mathrm{Inv}(f, M) = \left\{ x \in M : f^i(x) \in M, \text{ for } i \in \mathbb{Z} \right\}.$$

Then

$$f \text{ is strongly hyperbolic on } \mathcal{M} = \left(Q, \{(M_i, C_i)\}_{i=1}^N \right)$$

$$\Downarrow$$

$$f \text{ is uniformly hyperbolic on } \mathcal{H}$$

Strategy

1. **Step 1.** Enclose an invariant set.
2. **Step 2.** Detect (nonrigorously) periodic points in the invariant set up to some period.
3. **Step 3.** Set coordinate system at boxes that contain a k-periodic point x as a normalized Jordan basis of $Df^k(x)$.
4. **Step 4.** Propagate coordinate systems from periodic points to other boxes be means of the action of derivative of f.
5. **Step 5.** Verify strong hyperbolicity.

Strategy

1. **Step 1.** Enclose an invariant set.
2. **Step 2.** Detect (nonrigorously) periodic points in the invariant set up to some period.
3. **Step 3.** Set coordinate system at boxes that contain a k-periodic point x as a normalized Jordan basis of $Df^k(x)$.
4. **Step 4.** Propagate coordinate systems from periodic points to other boxes be means of the action of derivative of f.
5. **Step 5.** Verify strong hyperbolicity.

Strategy

1. **Step 1.** Enclose an invariant set.
2. **Step 2.** Detect (nonrigorously) periodic points in the invariant set up to some period.
3. **Step 3.** Set coordinate system at boxes that contain a k-periodic point x as a normalized Jordan basis of $Df^k(x)$.
4. **Step 4.** Propagate coordinate systems from periodic points to other boxes be means of the action of derivative of f.
5. **Step 5.** Verify strong hyperbolicity.

Strategy

1. **Step 1.** Enclose an invariant set.
2. **Step 2.** Detect (nonrigorously) periodic points in the invariant set up to some period.
3. **Step 3.** Set coordinate system at boxes that contain a k-periodic point x as a normalized Jordan basis of $Df^k(x)$.
4. **Step 4.** Propagate coordinate systems from periodic points to other boxes be means of the action of derivative of f.
5. **Step 5.** Verify strong hyperbolicity.

Verification of strong hyperbolicity

Strategy

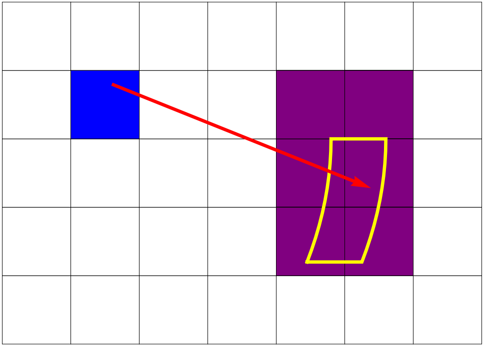
1. **Step 1.** Enclose an invariant set.
2. **Step 2.** Detect (nonrigorously) periodic points in the invariant set up to some period.
3. **Step 3.** Set coordinate system at boxes that contain a k-periodic point x as a normalized Jordan basis of $Df^k(x)$.
4. **Step 4.** Propagate coordinate systems from periodic points to other boxes be means of the action of derivative of f.
5. **Step 5.** Verify strong hyperbolicity.

Graph representation of maps.

$f\colon D \subset N \to N$ - a map
N_i - (usually) boxes that cover N

Directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a **graph representation** of f iff

$$f(N_i \cap D) \cap N_j \neq \emptyset \implies (i,j) \in \mathcal{E}$$



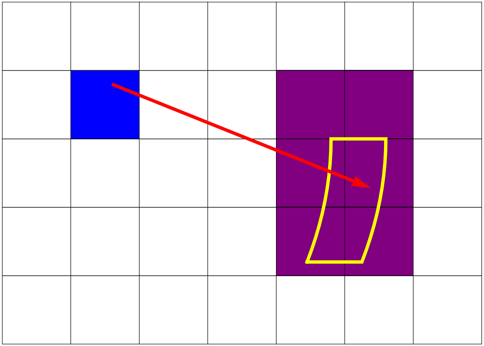
Clearly we can use interval arithmetic to compute graph representations of maps.

Graph representation of maps.

$f: D \subset N \to N$ - a map
N_i - (usually) boxes that cover N

Directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a **graph representation** of f iff

$$f(N_i \cap D) \cap N_j \neq \emptyset \implies (i,j) \in \mathcal{E}$$



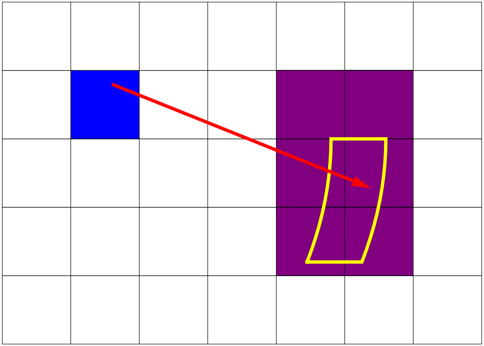
Clearly we can use interval arithmetic to compute graph representations of maps.

Graph representation of maps.

$f: D \subset N \to N$ - a map
N_i - (usually) boxes that cover N

Directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a **graph representation** of f iff

$$f(N_i \cap D) \cap N_j \neq \emptyset \Longrightarrow (i, j) \in \mathcal{E}$$



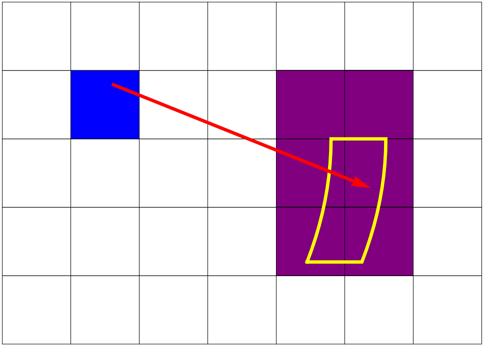
Clearly we can use interval arithmetic to compute graph representations of maps.

Graph representation of maps.

$f\colon D \subset N \to N$ - a map
N_i - (usually) boxes that cover N

Directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a **graph representation** of f iff

$$f(N_i \cap D) \cap N_j \neq \emptyset \implies (i, j) \in \mathcal{E}$$

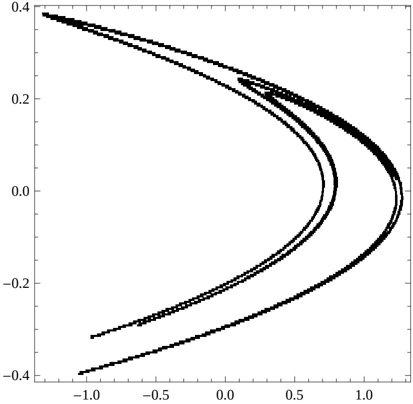


Clearly we can use interval arithmetic to compute graph representations of maps.

Enclosing attractors

"Inner" enclosure - good for attractors.

- take a box, say U, from the observed attracting domain
- compute $S_k = U \cup f(U) \cup \ldots \cup f^k(U)$ until $S_{k+1} = S_k$
- refine the graph, so that it does not contain vertexes without incoming edges



Enclosure of an invariant set for the Hénon map

$$\mathcal{H}(x,y) = (1 + y - ax^2, bx)$$

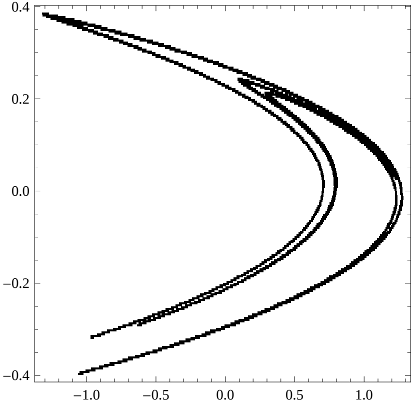
for the classical parameter values $a = 1.4$ and $b = 0.3$.

(Show the program)

Enclosing attractors

"Inner" enclosure - good for attractors.

- take a box, say U, from the observed attracting domain
- compute $S_k = U \cup f(U) \cup \ldots \cup f^k(U)$ until $S_{k+1} = S_k$
- refine the graph, so that it does not contain vertexes without incoming edges



Enclosure of an invariant set for the Hénon map

$$\mathcal{H}(x, y) = (1 + y - ax^2, bx)$$

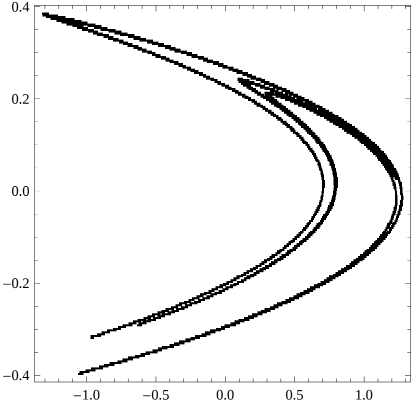
for the classical parameter values $a = 1.4$ and $b = 0.3$.

(Show the program)

Enclosing attractors

“Inner” enclosure - good for attractors.

- take a box, say U, from the observed attracting domain
- compute $S_k = U \cup f(U) \cup \ldots \cup f^k(U)$ until $S_{k+1} = S_k$
- refine the graph, so that it does not contain vertexes without incoming edges



Enclosure of an invariant set for the Hénon map

$$\mathcal{H}(x, y) = (1 + y - ax^2, bx)$$

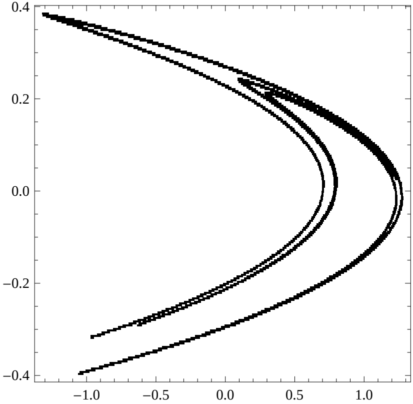
for the classical parameter values $a = 1.4$ and $b = 0.3$.

(Show the program)

Enclosing attractors

"Inner" enclosure - good for attractors.

- take a box, say U, from the observed attracting domain
- compute $S_k = U \cup f(U) \cup \ldots \cup f^k(U)$ until $S_{k+1} = S_k$
- refine the graph, so that it does not contain vertexes without incoming edges



Enclosure of an invariant set for the Hénon map

$$\mathcal{H}(x, y) = (1 + y - ax^2, bx)$$

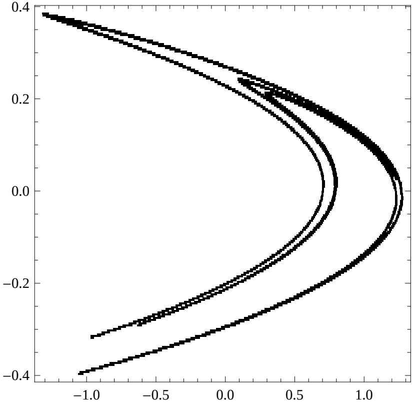
for the classical parameter values $a = 1.4$ and $b = 0.3$.

(Show the program)

Enclosing attractors

"Inner" enclosure - good for attractors.

- take a box, say U, from the observed attracting domain
- compute $S_k = U \cup f(U) \cup \ldots \cup f^k(U)$ until $S_{k+1} = S_k$
- refine the graph, so that it does not contain vertexes without incoming edges



Enclosure of an invariant set for the Hénon map

$$\mathcal{H}(x, y) = (1 + y - ax^2, bx)$$

for the classical parameter values $a = 1.4$ and $b = 0.3$.

(Show the program)

Coordinate systems at periodic points.

- Detect cycles in graph: all k-periodic points must belong to k-cycles.
- Using nonrigorous Newton method refine the cycles to periodic points.
- Compute coordinate systems at periodic points.

Coordinate systems at periodic points.

- Detect cycles in graph: all k-periodic points must belong to k-cycles.
- Using nonrigorous Newton method refine the cycles to periodic points.
- Compute coordinate systems at periodic points.
  - take x – k-periodic point
  - take M_i such that $x \in M_i$ and the coordinates C_i are not computed, yet
  - set C_i as the matrix of normalized eigenvectors of $Df^k(x)$. Columns of C_i are sorted by decreasing absolute value of the corresponding eigenvalue.

Coordinate systems at periodic points.

- Detect cycles in graph: all k-periodic points must belong to k-cycles.
- Using nonrigorous Newton method refine the cycles to periodic points.
- Compute coordinate systems at periodic points.
  - take x – k-periodic point
  - take M_i such that $x \in M_i$ and the coordinates C_i are not computed, yet
  - set C_i as the matrix of normalized eigenvectors of $Df^k(x)$. Columns of C_i are sorted by decreasing absolute value of the corresponding eigenvalue.

Coordinate systems at periodic points.

- Detect cycles in graph: all k-periodic points must belong to k-cycles.
- Using nonrigorous Newton method refine the cycles to periodic points.
- Compute coordinate systems at periodic points.
  - take x – k-periodic point
  - take M_i such that $x \in M_i$ and the coordinates C_i are not computed, yet
  - set C_i as the matrix of normalized eigenvectors of $Df^k(x)$. Columns of C_i are sorted by decreasing absolute value of the corresponding eigenvalue.

Coordinate systems at periodic points.

- Detect cycles in graph: all k-periodic points must belong to k-cycles.
- Using nonrigorous Newton method refine the cycles to periodic points.
- Compute coordinate systems at periodic points.
  - take x – k-periodic point
  - take M_i such that $x \in M_i$ and the coordinates C_i are not computed, yet
  - set C_i as the matrix of normalized eigenvectors of $Df^k(x)$. Columns of C_i are sorted by decreasing absolute value of the corresponding eigenvalue.

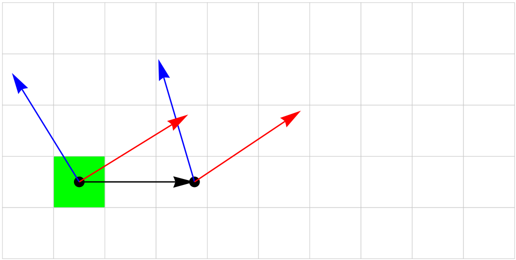
Coordinate systems at periodic points.

- Detect cycles in graph: all k-periodic points must belong to k-cycles.
- Using nonrigorous Newton method refine the cycles to periodic points.
- Compute coordinate systems at periodic points.
  - take x – k-periodic point
  - take M_i such that $x \in M_i$ and the coordinates C_i are not computed, yet
  - set C_i as the matrix of normalized eigenvectors of $Df^k(x)$. Columns of C_i are sorted by decreasing absolute value of the corresponding eigenvalue.

Spreading of the coordinates over the attractor

take the centre of a set M_i with computed coordinate system

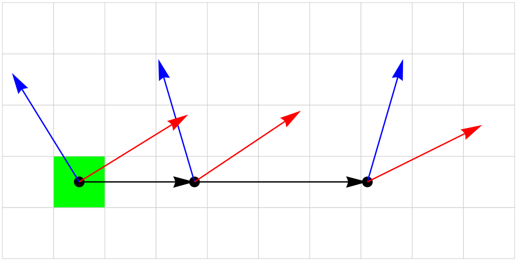
propagate coordinates by the action of Df



Spreading of the coordinates over the attractor

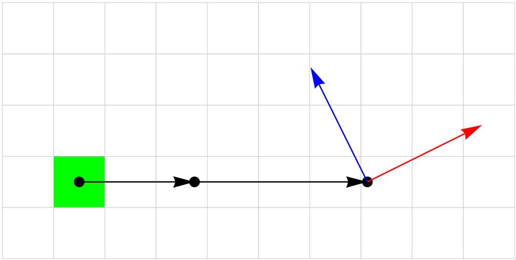
take the centre of a set M_i with computed coordinate system

propagate coordinates by the action of Df



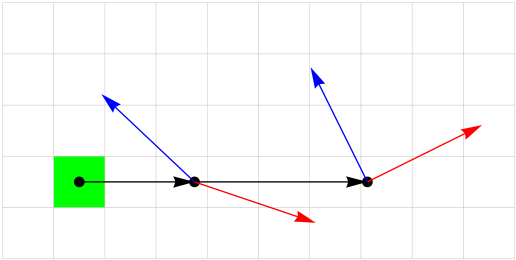
Spreading of the coordinates over the attractor

take the centre of a set M_i with computed coordinate system
propagate coordinates by the action of Df
orthonormalize after N steps



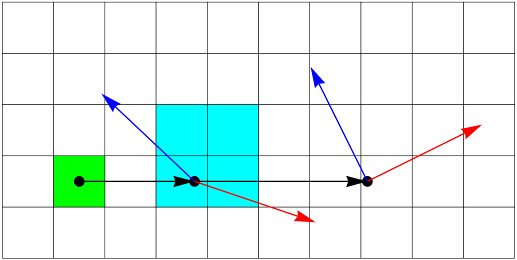
Spreading of the coordinates over the attractor

take the centre of a set M_i with computed coordinate system
propagate coordinates by the action of Df
orthonormalize after N steps
propagate these coordinates backwards $N - 1$ times



Spreading of the coordinates over the attractor

take the centre of a set M_i with computed coordinate system
propagate coordinates by the action of Df
orthonormalize after N steps
propagate these coordinates backwards $N-1$ times
normalize and set obtained coordinates to boxes
that are in the range of M_i



Spreading of the coordinates over the attractor

$N > 1$ - parameter

1. M_i - a set with computed C_i
2. $x := \mathrm{centre}(M_i)$
3. $x_k := f^k(x)$, $A_k := Df(x_k)$, $k = 0, 1, \ldots, N$
4. $A := A_N \cdots A_0 \cdot C_i$
5. $A := \mathrm{orthonormalize}(A)$
6. backward propagation: $C := (A_N \cdots A_1)^{-1} \cdot A$
7. normalize columns of C and set C as a coordinate system in each M_j such that $(i, j) \in \mathcal{E}$ and C_j is not computed

repeat (1-7) until C_i is computed for all M_i. It can be proved this procedure always stops if the graph has one strongly connected component

Spreading of the coordinates over the attractor

$N > 1$ - parameter

1. M_i - a set with computed C_i
2. $x := \mathrm{centre}(M_i)$
3. $x_k := f^k(x)$, $A_k := Df(x_k)$, $k = 0, 1, \ldots, N$

4. $A := A_N \cdots A_0 \cdot C_i$
5. $A := \mathrm{orthonormalize}(A)$
6. backward propagation: $C := (A_N \cdots A_1)^{-1} \cdot A$

7. normalize columns of C and set C as a coordinate system in each M_j such that $(i, j) \in \mathcal{E}$ and C_j is not computed

repeat (1-7) until C_i is computed for all M_i. It can be proved this procedure always stops if the graph has one strongly connected component

Spreading of the coordinates over the attractor

$N > 1$ - parameter

1. M_i - a set with computed C_i
2. $x := \mathrm{centre}(M_i)$
3. $x_k := f^k(x)$, $A_k := Df(x_k)$, $k = 0, 1, \ldots, N$

4. $A := A_N \cdots A_0 \cdot C_i$
5. $A := \mathrm{orthonormalize}(A)$
6. backward propagation: $C := (A_N \cdots A_1)^{-1} \cdot A$

7. normalize columns of C and set C as a coordinate system in each M_j such that $(i, j) \in \mathcal{E}$ and C_j is not computed

repeat (1-7) until C_i is computed for all M_i. It can be proved this procedure always stops if the graph has one strongly connected component

Spreading of the coordinates over the attractor

$N > 1$ - parameter

1. M_i - a set with computed C_i
2. $x := \mathrm{centre}(M_i)$
3. $x_k := f^k(x)$, $A_k := Df(x_k)$, $k = 0, 1, \ldots, N$

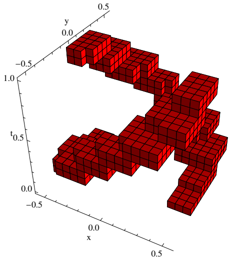
4. $A := A_N \cdots A_0 \cdot C_i$
5. $A := \mathrm{orthonormalize}(A)$
6. backward propagation: $C := (A_N \cdots A_1)^{-1} \cdot A$

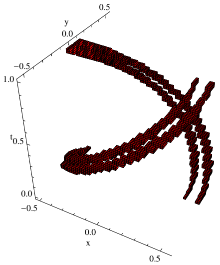
7. normalize columns of C and set C as a coordinate system in each M_j such that $(i, j) \in \mathcal{E}$ and C_j is not computed

repeat (1-7) until C_i is computed for all M_i. It can be proved this procedure always stops if the graph has one strongly connected component

Test case: Smale map

$$s(x, y, t) = (0.1x + 0.5\cos(2\pi t), 0.1y + 0.5\sin(2\pi t), 2t \bmod 1).$$





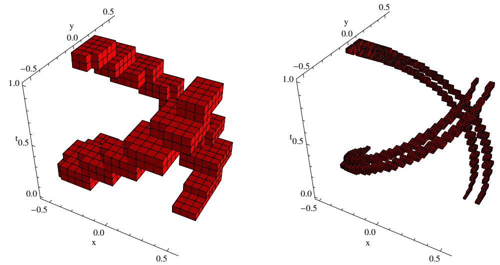
Found 1 candidate for fixed point, a period 2 orbit and two period 3 orbits.
Spreading of coordinate systems with $N = 2$.
Verification of the strong hyperbolicity with

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

All the computations took less than 1 second.

Test case: Smale map

$$s(x, y, t) = (0.1x + 0.5\cos(2\pi t), 0.1y + 0.5\sin(2\pi t), 2t \bmod 1).$$



Found 1 candidate for fixed point, a period 2 orbit and two period 3 orbits.

Spreading of coordinate systems with $N = 2$.

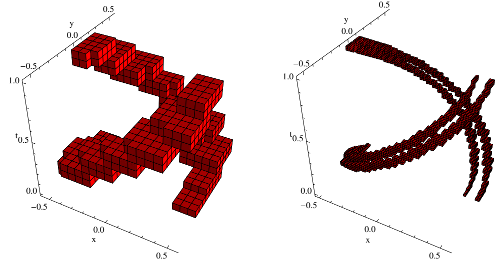
Verification of the strong hyperbolicity with

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

All the computations took less than 1 second.

Test case: Smale map

$$s(x, y, t) = (0.1x + 0.5\cos(2\pi t), 0.1y + 0.5\sin(2\pi t), 2t \bmod 1).$$



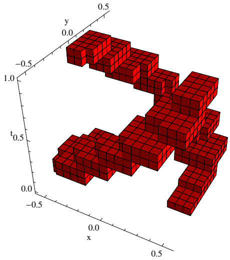
Found 1 candidate for fixed point, a period 2 orbit and two period 3 orbits.
Spreading of coordinate systems with $N = 2$.
Verification of the strong hyperbolicity with

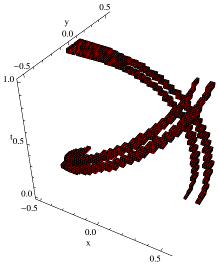
$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

All the computations took less than 1 second.

Test case: Smale map

$$s(x, y, t) = (0.1x + 0.5\cos(2\pi t), 0.1y + 0.5\sin(2\pi t), 2t \bmod 1).$$





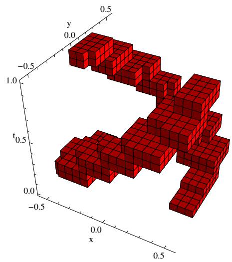
Found 1 candidate for fixed point, a period 2 orbit and two period 3 orbits.
Spreading of coordinate systems with $N = 2$.
Verification of the strong hyperbolicity with

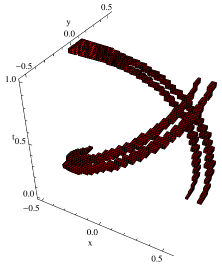
$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

All the computations took less than 1 second.

Test case: Smale map

$$s(x, y, t) = (0.1x + 0.5\cos(2\pi t), 0.1y + 0.5\sin(2\pi t), 2t \bmod 1).$$





Found 1 candidate for fixed point, a period 2 orbit and two period 3 orbits.
Spreading of coordinate systems with $N = 2$.
Verification of the strong hyperbolicity with

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

All the computations took less than 1 second.

Verification of hyperbolicity in the Kuznetsov system

algorithm	wall time (h:mm)	comments
enclosure of attractor	2:16 on 224 CPUs	7 970 392 boxes
cycles in graph, max period 6	0:58 on 32 CPUs	2190 cycles found
periodic points, max period 6	0:06 on 32 CPUs	105 points found
coordinate systems	6:59 on 32 CPUs	parameter $N = 2$
strong hyperbolicity	4:24 on 224 CPUs	integration of $\mathcal{C}^1$ computations

Quadratic form:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).
Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Periodic orbits in the attractor

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).
Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Periodic orbits in the attractor

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).

Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Periodic orbits in the attractor

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).
Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).
Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Periodic orbits in the attractor

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).
Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Periodic orbits in the attractor

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).

Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Periodic orbits in the attractor

Why the attractor is nontrivial?

- $\mathcal{B}$ - set of boxes that cover the attractor and $P(\mathcal{B}) \subset \mathcal{B}$.
- $\mathcal{B}$ is compact connected - we computed homology groups of the set
- therefore $\mathcal{A} = \bigcup_{n>0} P^n(\mathcal{B})$ is compact and connected

It is enough to show that it contains two different points.

Main tool: the Interval Newton operator (integration of variational equations).
Fixed point – solve

$$P(x) - x = 0.$$

Period two point – solve

$$(P(x) - y, P(y) - x) = (0, 0), \quad x \neq y.$$

Good approximations for periodic points already computed from the previous steps. The above solved with the accuracy 10^{-12}.

Open problem

Plykin type attractor - (Kuznetsov 2009).

$$
\begin{aligned}
\dot{x} &= -2\varepsilon y^2\Omega_1(x,y,t)\left(\cos\left(\tfrac{\pi}{4}\cos\tfrac{\pi}{2}t\right) - x\sin\left(\tfrac{\pi}{4}\cos\tfrac{\pi}{2}t\right)\right) + \\
&\quad Ky\Omega_2(x,y,t)\left(\cos\left(\tfrac{\pi}{4}\sin\tfrac{\pi}{2}t\right) - x\sin\left(\tfrac{\pi}{4}\sin\tfrac{\pi}{2}t\right)\right)\sin\tfrac{\pi}{2}t, \\
\dot{y} &= \varepsilon y\Omega_1(x,y,t)\left(2x\cos\left(\tfrac{\pi}{4}\cos\tfrac{\pi}{2}t\right) + (1-x^2+y^2)\sin\left(\tfrac{\pi}{4}\cos\tfrac{\pi}{2}t\right)\right) - \\
&\quad K\Omega_2(x,y,t)\left(x\cos\left(\tfrac{\pi}{4}\sin\tfrac{\pi}{2}t\right) + \tfrac{1}{2}(1-x^2+y^2)\sin\left(\tfrac{\pi}{4}\sin\tfrac{\pi}{2}t\right)\right)\sin\tfrac{\pi}{2}t,
\end{aligned}
$$

where

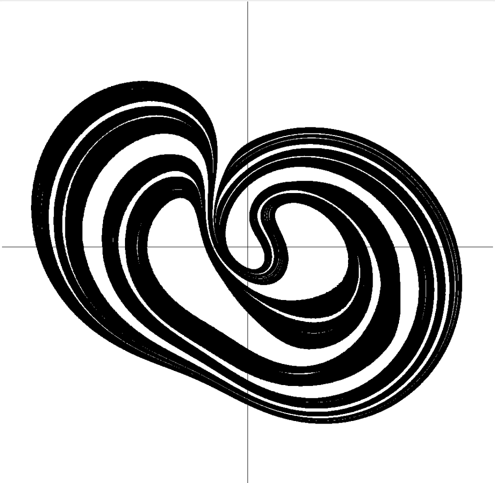
$$
\begin{aligned}
\Omega_1(x,y,t) &= \frac{2x\cos\left(\frac{\pi}{4}\cos\frac{\pi}{2}t\right)+(1-x^2-y^2)\sin\left(\frac{\pi}{4}\cos\frac{\pi}{2}t\right)}{\left(1+x^2+y^2\right)^2}, \\
\Omega_2(x,y,t) &= \frac{-2x\sin\left(\frac{\pi}{4}\sin\frac{\pi}{2}t\right)+(1-x^2-y^2)\cos\left(\frac{\pi}{4}\sin\frac{\pi}{2}t\right)}{\left(1+x^2+y^2\right)^2} + \frac{1}{\sqrt{2}}.
\end{aligned}
$$

S.P. Kuznetsov, A non-autonomous flow system with Plykin type attractor,
Communications in Nonlinear Science and Numerical Simulation, 14, 2009, 3487–3491

Partial result

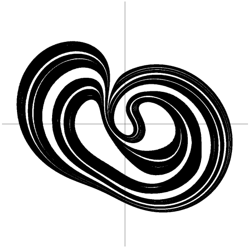
Theorem

For $K = 1.9$, $\varepsilon = 0.72$ there is a compact, connected set $\mathcal{B}$ such that the Poincaré map defined as a shift along the trajectories over the period of the vector field is positive invariant on $\mathcal{B}$.



Open problem

- Nonrigorous simulation strongly suggests the attractor is hyperbolic.
- We were not able to verify hyperbolicity due to memory limitations.



Click here to start animation

taken from

http://www.sgtnd.narod.ru/science/hyper/eng/index.htm

$\mathcal{C}^r$ Solvers

Higher order variational equation

Problem to solve:

$$x'(t) = f(x(t)) \quad - \quad \text{main ODE}$$

Compute:

$$\begin{aligned} x(t) &= \varphi(t,x) \\ V(t) &= D_x\varphi(t,x) \\ H(t) &= D_{x,x}\varphi(t,x) \\ \ldots \end{aligned}$$

Initial conditions:

$$\begin{aligned} x(0) &\in [X] \\ V(0) &\in [V] \qquad \text{often } [V] = \{\mathrm{Id}\} \\ H(0) &\in [H] \qquad \text{often } [H] = 0 \\ \ldots \end{aligned}$$

Higher order variational equation

Problem to solve:

$$x'(t) = f(x(t)) \quad - \quad \text{main ODE}$$

Compute:

$$\begin{array}{rcl} x(t) & = & \varphi(t,x) \\ V(t) & = & D_x\varphi(t,x) \\ H(t) & = & D_{x,x}\varphi(t,x) \\ & \dots & \end{array}$$

Initial conditions:

$$\begin{array}{rcll} x(0) & \in & [X] & \\ V(0) & \in & [V] & \text{often } [V] = \{\mathrm{Id}\} \\ H(0) & \in & [H] & \text{often } [H] = 0 \\ & \dots & & \end{array}$$

Higher order variational equation

Problem to solve:

$$x'(t) = f(x(t)) \quad - \quad \text{main ODE}$$

Compute:

$$\begin{aligned} x(t) &= \varphi(t, x) \\ V(t) &= D_x\varphi(t, x) \\ H(t) &= D_{x,x}\varphi(t, x) \\ &\dots \end{aligned}$$

Initial conditions:

$$\begin{aligned} x(0) &\in [X] & & \\ V(0) &\in [V] & & \text{often } [V] = \{\mathrm{Id}\} \\ H(0) &\in [H] & & \text{often } [H] = 0 \\ &\dots \end{aligned}$$

Structure of variational equation

Fact: a - multiindex

$$\frac{d}{dt} D_a\varphi(t,x) = Df(x(t)) \cdot D_a\varphi(t,x) + l.o.t.$$

is nonautonomous linear.

- It is enough to compute

  $$D_a\varphi(h,[X])$$

  where h is the time step.
- similar strategy for propagation of products as in $\mathcal{C}^1$ algorithm.

DW, P. Zgliczyński, $\mathcal{C}^r$-Lohner algorithm, Schedae Informaticae, (2011), 20, 9-46.

Structure of variational equation

Fact: a - multiindex

$$\frac{d}{dt} D_a \varphi(t, x) = Df(x(t)) \cdot D_a \varphi(t, x) + l.o.t.$$

is nonautonomous linear.

- It is enough to compute

$$D_a \varphi(h, [X])$$

  where h is the time step.

- similar strategy for propagation of products as in $\mathcal{C}^1$ algorithm.

DW, P. Zgliczyński, $\mathcal{C}^r$-Lohner algorithm, Schedae Informaticae, (2011), 20, 9-46.

Structure of variational equation

Fact: a - multiindex

$$\frac{d}{dt} D_a\varphi(t,x) = Df(x(t)) \cdot D_a\varphi(t,x) + l.o.t.$$

is nonautonomous linear.

- It is enough to compute

  $$D_a\varphi(h,[X])$$

  where h is the time step.
- similar strategy for propagation of products as in $\mathcal{C}^1$ algorithm.

DW, P. Zgliczyński, $\mathcal{C}^r$-Lohner algorithm, Schedae Informaticae, (2011), 20, 9-46.

Structure of variational equation

Fact: a - multiindex

$$\frac{d}{dt}D_a\varphi(t,x) = Df(x(t)) \cdot D_a\varphi(t,x) + l.o.t.$$

is nonautonomous linear.

- It is enough to compute

$$D_a\varphi(h,[X])$$

  where h is the time step.
- similar strategy for propagation of products as in $\mathcal{C}^1$ algorithm.

DW, P. Zgliczyński, $\mathcal{C}^r$-Lohner algorithm, Schedae Informaticae, (2011), 20, 9-46.

```cpp
/** Integration of higher order variational equations **/
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;
int main(){
  // last argument specifies maximal derivative
  const int degree = 4;
  IMap pendulum("time:t;var:x,y;fun:y,-sin(x);",degree);
  ICnOdeSolver solver(pendulum,20);  // ODE integrator
  ICnTimeMap tm(solver);             // class for long time integration

  IVector u({1,0});
  // representation of initial condition,
  // initial condition for first order equations is Id by default
  // initial condition for higher order equations is zero by default
  CnRect2Set s(u,degree);
  // integrate until T=2 and print result
  cout << "phi(2,u)=" << tm(2.,s) << endl;
  // print one particular derivative
  cout << "D_{x,x,y}phi_y(t,u)=" << s(1,Multipointer{0,0,1}) << endl;
  // print vector of derivatives
  cout << "D_{x,x,y}phi(t,u)=" << s(Multipointer{0,0,1}) << endl;
  // print all derivatives
  cout << s.currentSet().toString();
  return 0;
}
```

```
/* Output of the program:
phi(2,u)={[-0.184719, -0.184719],[-0.899896, -0.899896]}
D_{x,x,y}phi_y(t,u)=[0.170001, 0.170001]
D_{x,x,y}phi(t,u)={[0.275761, 0.275761],[0.170001, 0.170001]}

value :
   {0,0}  : {[-0.1847185233641392, -0.1847185233641322],[-0.8998955756501549, -0.8998955756501478]}
Taylor coefficients of order 1 :
   {1,0}  : {[-0.06176642316703151, -0.06176642316701867],[-0.8893742085818004, -0.8893742085817878]}
   {0,1}  : {[1.100408851124127, 1.100408851124144],[-0.3452482404138276, -0.3452482404138115]}
Taylor coefficients of order 2 :
   {2,0}  : {[0.3689193486997759, 0.3689193486997865],[0.04249736468215068, 0.04249736468215968]}
   {1,1}  : {[0.4033277498641553, 0.4033277498641712],[0.1690697521607405, 0.1690697521607553]}
   {0,2}  : {[0.1918365535108403, 0.1918365535108501],[0.1289933705787395, 0.1289933705787497]}
Taylor coefficients of order 3 :
   {3,0}  : {[0.150665473381, 0.150665473381008],[0.05124435888808759, 0.05124435888809471]}
   {2,1}  : {[0.2757613878113712, 0.2757613878113923],[0.1700006453873801, 0.1700006453874003]}
   {1,2}  : {[0.2439399987249618, 0.2439399987249848],[0.2193554100897172, 0.2193554100897407]}
   {0,3}  : {[0.1213946818536904, 0.1213946818537009],[0.2022733384746434, 0.2022733384746553]}
Taylor coefficients of order 4 :
   {4,0}  : {[0.0250738578068196, 0.02507385780682729],[0.03541789401056386, 0.03541789401057061]}
   {3,1}  : {[0.09187471323467304, 0.09187471323470189],[0.1229862385399809, 0.1229862385400072]}
   {2,2}  : {[0.1195564021283123, 0.1195564021283555],[0.2380738082223464, 0.2380738082223872]}
   {1,3}  : {[0.06018389101213596, 0.06018389101216642],[0.1614399291087914, 0.1614399291088211]}
   {0,4}  : {[0.01389092366608679, 0.01389092366609591],[0.05235840754075258, 0.05235840754076206]}
}*/
```

```cpp
/** Higher order derivatives of Poincare map **/
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
using namespace std;

int main()
{
  // Instance of the vector field, ODE solver, Poincare map
  int degree = 4;
  IMap vf("var:x,y;fun:y,(1-x^2)*y-x;",degree);
  ICnOdeSolver solver(vf, 20);
  ICoordinateSection section(2, 1);
  ICnPoincareMap pm(solver, section, poincare::PlusMinus);

  // Take a ball centred at approximate periodic point
  IVector u({2.0086198608748433,0.});
  CnRect2Set s(u,degree);
  // data structure to store Taylor coefficients
  IJet jet(2,degree);
  // Call routine that computes rigorously Poincare map
  IVector y = pm(s,jet);
  // recompute Taylor coeffs of flow to Taylor coeffs of P. Map
  jet = pm.computeDP(jet);
  // print derivatives
  cout << jet.toString();
  return 0;
}
```

```
/*Output:

value :
   {0,0}  : {[2.008619860874834, 2.008619860874853],[-1.674225432758842e-13, 1.674225432758742e-13]}
Taylor coefficients of order 1 :
   {1,0}  : {[0.0008596950592716135, 0.0008596950679375556],[-1.204325528192385e-11, 1.20436993711337e-
   {0,1}  : {[-1.593917860307063e-12, 1.594439786883827e-12],[-4.428124533717437e-12, 4.42801351141497
Taylor coefficients of order 2 :
   {2,0}  : {[-0.002296951610312573, -0.00229695134283375],[-4.308036150035832e-10, 4.308036150035832e
   {1,1}  : {[-9.810905124258537e-11, 9.81092732871903e-11],[-3.156544470250822e-10, 3.15654447025082
   {0,2}  : {[0.0002140014165422138, 0.0002140014526664741],[-5.804126623765171e-11, 5.80412662376517
Taylor coefficients of order 3 :
   {3,0}  : {[0.002994126624442359, 0.002994132727242985],[-1.11234532695903e-08, 1.11234532695903e-0
   {2,1}  : {[-3.353211717694936e-09, 3.353195286394172e-09],[-1.219278639164045e-08, 1.21927863638848
   {1,2}  : {[-0.00125008989286738 9, -0.001250087427980784],[-4.471445103071403e-09, 4.47144510307140
   {0,3}  : {[-0.0002155374854198236, -0.0002155371826957891],[-5.482866533475702e-10, 5.482866545040
Taylor coefficients of order 4 :
   {4,0}  : {[-0.001681475844898417, -0.001681360601371844],[-2.323861542456266e-07, 2.32386154282634
   {3,1}  : {[-8.434543574425733e-08, 8.434541561221314e-08],[-3.390407211645297e-07, 3.3904072116452
   {2,2}  : {[0.002858275980980476, 0.002858368851661084],[-1.861387415591498e-07, 1.861387415591498e
   {1,3}  : {[0.001081020162548411, 0.0010810429386651],[-4.554923604960237e-08, 4.554923604960237e-0
   {0,4}  : {[8.862824145262416e-05, 8.863033955809182e-05],[-4.189210744502218e-09, 4.18921074681518
*/
```

Applications

Elliptic orbits and invariant tori

Linear stability in Hamiltonan systems

$$\text{linear stability of periodic solutions } t \to u(t)$$
$$\Updownarrow$$
$$\text{linear stability of fixed point } u_0 \text{ of a Poincaré map } P$$
$$\Updownarrow$$
$$\sigma(DP(u_0)) \subset \mathcal{S}^1$$

Planar case: eigenvalues of $DP(u_0)$ are

- either real
- or complex conjugated

Enough to check: $\det(DP(u_0) - \lambda \mathrm{Id}) = 0$ has no real solutions.

Fact:

Linear stability does not imply stability

Linear stability in Hamiltonan systems

$$\text{linear stability of periodic solutions } t \to u(t)$$
$$\Updownarrow$$
$$\text{linear stability of fixed point } u_0 \text{ of a Poincaré map } P$$
$$\Updownarrow$$
$$\sigma(DP(u_0)) \subset \mathcal{S}^1$$

Planar case: eigenvalues of $DP(u_0)$ are

- either real
- or complex conjugated

Enough to check: $\det(DP(u_0) - \lambda \mathrm{Id}) = 0$ has no real solutions.

Fact:

Linear stability does not imply stability

Linear stability in Hamiltonan systems

$$\text{linear stability of periodic solutions } t \to u(t)$$
$$\Updownarrow$$
$$\text{linear stability of fixed point } u_0 \text{ of a Poincaré map } P$$
$$\Updownarrow$$
$$\sigma(DP(u_0)) \subset \mathcal{S}^1$$

Planar case: eigenvalues of $DP(u_0)$ are

- either real
- or complex conjugated

Enough to check: $\det(DP(u_0) - \lambda \mathrm{Id}) = 0$ has no real solutions.

Fact:

Linear stability does not imply stability

Theoretical tool

Theorem (Moser)

Consider an analytic area preserving map $f : \mathbb{R}^2 \to \mathbb{R}^2$, $f(r, s) = (r_1, s_1)$ where

$$\begin{aligned} r_1 &= r \cos\alpha - s \sin\alpha + O_{2l+2}, \\ s_1 &= r \sin\alpha + s \cos\alpha + O_{2l+2}, \\ \alpha &= \sum_{k=0}^{l} \gamma_k \left(r^2 + s^2\right)^k \end{aligned}$$

and O_{2l+2} denotes convergent power series in r, s with terms of order greater than $2l + 1$, only.

If at least one of $\gamma_1, \ldots, \gamma_l$ is not zero then the origin is a stable fixed point for f. Moreover, in any neighborhood U containing zero there exists an invariant curve for f around the origin contained in U.

Theoretical tool

Theorem (Moser)

Consider an analytic area preserving map $f : \mathbb{R}^2 \to \mathbb{R}^2$, $f(r,s) = (r_1, s_1)$ where

$$\begin{aligned} r_1 &= r\cos\alpha - s\sin\alpha + O_{2l+2}, \\ s_1 &= r\sin\alpha + s\cos\alpha + O_{2l+2}, \\ \alpha &= \sum_{k=0}^{l} \gamma_k \left(r^2 + s^2\right)^k \end{aligned}$$

and O_{2l+2} denotes convergent power series in r, s with terms of order greater than $2l+1$, only.

If at least one of $\gamma_1, \ldots, \gamma_l$ is not zero then the origin is a stable fixed point for f. Moreover, in any neighborhood U containing zero there exists an invariant curve for f around the origin contained in U.

Birkhoff normal form

Algorithm for computing Birkhoff normal form:

Input:

- truncated complex polynomial of an area preserving map

$$f(x,y) = (\lambda x, \overline{\lambda} y) + (f_{2,x}(x,y), f_{2,y}(x,y)) + h.o.t.$$

- linear part is elliptic, i.e. $|\lambda| = 1$
- no strong resonances ($\lambda^i \neq 1$ for $i = 1,2,3,4$)

Output:

- explicit and area preserving change of coordinates

$$(x,y) = (z,w) + (C_{2,x}(z,w), C_{2,y}(z,w)) + h.o.t.$$

- that brings f to its (truncated) normal form

$$F(z,w) = \left(z e^{i(\gamma_0 + \gamma_1 zw)}, w e^{-i(\gamma_0 + \gamma_1 zw)} \right) + h.o.t.$$

Birkhoff normal form

Algorithm for computing Birkhoff normal form:

Input:

- truncated complex polynomial of an area preserving map

$$f(x,y) = (\lambda x, \overline{\lambda} y) + (f_{2,x}(x,y), f_{2,y}(x,y)) + h.o.t.$$

- linear part is elliptic, i.e. $|\lambda| = 1$
- no strong resonances ($\lambda^i \neq 1$ for $i = 1,2,3,4$)

Output:

- explicit and area preserving change of coordinates

$$(x,y) = (z,w) + (C_{2,x}(z,w), C_{2,y}(z,w)) + h.o.t.$$

- that brings f to its (truncated) normal form

$$F(z,w) = \left(z e^{i(\gamma_0 + \gamma_1 zw)}, w e^{-i(\gamma_0 + \gamma_1 zw)}\right) + h.o.t.$$

Birkhoff normal form

Algorithm for computing Birkhoff normal form:

Input:

- truncated complex polynomial of an area preserving map

$$f(x,y) = (\lambda x, \overline{\lambda} y) + (f_{2,x}(x,y), f_{2,y}(x,y)) + h.o.t.$$

- linear part is elliptic, i.e. $|\lambda| = 1$
- no strong resonances ($\lambda^i \neq 1$ for $i = 1,2,3,4$)

Output:

- explicit and area preserving change of coordinates

$$(x,y) = (z,w) + (C_{2,x}(z,w), C_{2,y}(z,w)) + h.o.t.$$

- that brings f to its (truncated) normal form

$$F(z,w) = \left(z e^{i(\gamma_0 + \gamma_1 zw)}, w e^{-i(\gamma_0 + \gamma_1 zw)} \right) + h.o.t.$$

Algorithm for computing Birkhoff normal form:

Input:

- truncated complex polynomial of an area preserving map

$$f(x,y) = (\lambda x, \overline{\lambda} y) + (f_{2,x}(x,y), f_{2,y}(x,y)) + h.o.t.$$

- linear part is elliptic, i.e. $|\lambda| = 1$
- no strong resonances ($\lambda^i \neq 1$ for $i = 1,2,3,4$)

Output:

- explicit and area preserving change of coordinates

$$(x,y) = (z,w) + (C_{2,x}(z,w), C_{2,y}(z,w)) + h.o.t.$$

- that brings f to its (truncated) normal form

$$F(z,w) = \left(z e^{i(\gamma_0 + \gamma_1 zw)}, w e^{-i(\gamma_0 + \gamma_1 zw)} \right) + h.o.t.$$

Algorithm for computing Birkhoff normal form:

Input:

- truncated complex polynomial of an area preserving map

$$f(x, y) = (\lambda x, \overline{\lambda} y) + (f_{2,x}(x, y), f_{2,y}(x, y)) + h.o.t.$$

- linear part is elliptic, i.e. $|\lambda| = 1$
- no strong resonances ($\lambda^i \neq 1$ for $i = 1, 2, 3, 4$)

Output:

- explicit and area preserving change of coordinates

$$(x, y) = (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + h.o.t.$$

- that brings f to its (truncated) normal form

$$F(z, w) = \left(ze^{i(\gamma_0+\gamma_1 zw)}, we^{-i(\gamma_0+\gamma_1 zw)}\right) + h.o.t.$$

Change of variables

$$(x, y) = C(z, w) := (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + \cdots$$

Solve functional equation

$$f(C(z, w)) = C(F(z, w))$$

Comparing first order derivatives we get

$$D(f \circ C)(0, 0) = DiagMatrix(\lambda, \overline{\lambda})$$

and

$$D(C \circ F)(0, 0) = DiagMatrix\left(e^{i\gamma_0}, e^{-i\gamma_0}\right)$$

Thus

$$\gamma_0 = \mathrm{Arg}\lambda$$

Homological equation

Change of variables

$$(x, y) = C(z, w) := (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + \cdots$$

Solve functional equation

$$f(C(z, w)) = C(F(z, w))$$

Comparing first order derivatives we get

$$D(f \circ C)(0, 0) = DiagMatrix(\lambda, \overline{\lambda})$$

and

$$D(C \circ F)(0, 0) = DiagMatrix\left(e^{i\gamma_0}, e^{-i\gamma_0}\right)$$

Thus

$$\gamma_0 = \mathrm{Arg}\lambda$$

Homological equation

Change of variables

$$(x, y) = C(z, w) := (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + \cdots$$

Solve functional equation

$$f(C(z, w)) = C(F(z, w))$$

Comparing first order derivatives we get

$$D(f \circ C)(0, 0) = DiagMatrix(\lambda, \overline{\lambda})$$

and

$$D(C \circ F)(0, 0) = DiagMatrix\left(e^{i\gamma_0}, e^{-i\gamma_0}\right)$$

Thus

$$\gamma_0 = \mathrm{Arg}\lambda$$

Homological equation

Change of variables

$$(x, y) = C(z, w) := (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + \cdots$$

Solve functional equation

$$f(C(z, w)) = C(F(z, w))$$

Comparing first order derivatives we get

$$D(f \circ C)(0, 0) = DiagMatrix(\lambda, \overline{\lambda})$$

and

$$D(C \circ F)(0, 0) = DiagMatrix\left(e^{i\gamma_0}, e^{-i\gamma_0}\right)$$

Thus

$$\gamma_0 = \mathrm{Arg}\lambda$$

Change of variables

$$(x, y) = C(z, w) := (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + \cdots$$

Solve functional equation

$$f(C(z, w)) = C(F(z, w))$$

Comparing first order derivatives we get

$$D(f \circ C)(0, 0) = DiagMatrix(\lambda, \overline{\lambda})$$

and

$$D(C \circ F)(0, 0) = DiagMatrix\left(e^{i\gamma_0}, e^{-i\gamma_0}\right)$$

Thus

$$\gamma_0 = \mathrm{Arg}\lambda$$

Homological equation

Change of variables

$$(x, y) = C(z, w) := (z, w) + (C_{2,x}(z, w), C_{2,y}(z, w)) + \cdots$$

Solve functional equation

$$f(C(z, w)) = C(F(z, w))$$

Comparing first order derivatives we get

$$D(f \circ C)(0, 0) = DiagMatrix(\lambda, \overline{\lambda})$$

and

$$D(C \circ F)(0, 0) = DiagMatrix\left(e^{i\gamma_0}, e^{-i\gamma_0}\right)$$

Thus

$$\gamma_0 = \mathrm{Arg}\lambda$$

Homological equation

Solve for $C_2 = (C_{2,x}, C_{2,y})$:

(second degree homogeneous polynomials are identified with symmetric matrices)

$$D^2(f_x \circ C) = f_{2,x} + \lambda C_{2,x}$$

and

$$D^2(C_x \circ F) = F_{2,x} (= 0) + \begin{bmatrix} \lambda^2 (C_{2,x})_{11} & (C_{2,x})_{12} \\ (C_{2,x})_{21} & \overline{\lambda}^2 C_{2,x})_{22} \end{bmatrix}$$

This leads to

$$\begin{aligned} (\lambda^2 - \lambda)(C_{2,x})_{11} &= (f_{2,x})_{11} \\ (\overline{\lambda}^2 - \lambda)(C_{2,x})_{22} &= (f_{2,x})_{22} \\ (1 - \lambda)(C_{2,x})_{12} &= (f_{2,x})_{12} \end{aligned}$$

Strong resonances

This is why we have to assume $\lambda^i \neq 1$.

Homological equation

Solve for $C_2 = (C_{2,x}, C_{2,y})$:

(second degree homogeneous polynomials are identified with symmetric matrices)

$$D^2(f_x \circ C) = f_{2,x} + \lambda C_{2,x}$$

and

$$D^2(C_x \circ F) = F_{2,x} (= 0) + \begin{bmatrix} \lambda^2 (C_{2,x})_{11} & (C_{2,x})_{12} \\ (C_{2,x})_{21} & \bar{\lambda}^2 C_{2,x})_{22} \end{bmatrix}$$

This leads to

$$\begin{aligned} (\lambda^2 - \lambda)(C_{2,x})_{11} &= (f_{2,x})_{11} \\ (\bar{\lambda}^2 - \lambda)(C_{2,x})_{22} &= (f_{2,x})_{22} \\ (1 - \lambda)(C_{2,x})_{12} &= (f_{2,x})_{12} \end{aligned}$$

Strong resonances

This is why we have to assume $\lambda^i \neq 1$.

Homological equation

Solve for $C_2 = (C_{2,x}, C_{2,y})$:

(second degree homogeneous polynomials are identified with symmetric matrices)

$$D^2(f_x \circ C) = f_{2,x} + \lambda C_{2,x}$$

and

$$D^2(C_x \circ F) = F_{2,x}(= 0) + \begin{bmatrix} \lambda^2 (C_{2,x})_{11} & (C_{2,x})_{12} \\ (C_{2,x})_{21} & \bar{\lambda}^2 C_{2,x})_{22} \end{bmatrix}$$

This leads to

$$\begin{aligned} (\lambda^2 - \lambda)(C_{2,x})_{11} &= (f_{2,x})_{11} \\ (\bar{\lambda}^2 - \lambda)(C_{2,x})_{22} &= (f_{2,x})_{22} \\ (1 - \lambda)(C_{2,x})_{12} &= (f_{2,x})_{12} \end{aligned}$$

Strong resonances

This is why we have to assume $\lambda^i \neq 1$.

Homological equation

Solve for $C_2 = (C_{2,x}, C_{2,y})$:

(second degree homogeneous polynomials are identified with symmetric matrices)

$$D^2(f_x \circ C) = f_{2,x} + \lambda C_{2,x}$$

and

$$D^2(C_x \circ F) = F_{2,x}(= 0) + \begin{bmatrix} \lambda^2 (C_{2,x})_{11} & (C_{2,x})_{12} \\ (C_{2,x})_{21} & \bar{\lambda}^2 C_{2,x})_{22} \end{bmatrix}$$

This leads to

$$\begin{aligned} (\lambda^2 - \lambda)(C_{2,x})_{11} &= (f_{2,x})_{11} \\ (\bar{\lambda}^2 - \lambda)(C_{2,x})_{22} &= (f_{2,x})_{22} \\ (1 - \lambda)(C_{2,x})_{12} &= (f_{2,x})_{12} \end{aligned}$$

Strong resonances

This is why we have to assume $\lambda^i \neq 1$.

Homological equation

Solve for $C_2 = (C_{2,x}, C_{2,y})$:

(second degree homogeneous polynomials are identified with symmetric matrices)

$$D^2(f_x \circ C) = f_{2,x} + \lambda C_{2,x}$$

and

$$D^2(C_x \circ F) = F_{2,x}(= 0) + \begin{bmatrix} \lambda^2 (C_{2,x})_{11} & (C_{2,x})_{12} \\ (C_{2,x})_{21} & \bar{\lambda}^2 C_{2,x})_{22} \end{bmatrix}$$

This leads to

$$\begin{aligned} (\lambda^2 - \lambda)(C_{2,x})_{11} &= (f_{2,x})_{11} \\ (\bar{\lambda}^2 - \lambda)(C_{2,x})_{22} &= (f_{2,x})_{22} \\ (1 - \lambda)(C_{2,x})_{12} &= (f_{2,x})_{12} \end{aligned}$$

Strong resonances

This is why we have to assume $\lambda^i \neq 1$.

Homological equation

Solve $C_3 = (C_{3,x}, C_{3,y})$ comparing third order terms

Here we have an extra variable γ_1 which makes the solution not unique.

Reality condition

$\gamma_1 \in \mathbb{R}$ makes the solution unique.

Explicit (not very complicated:-) formulae for C_3 and γ_1

Homological equation

Solve $C_3 = (C_{3,x}, C_{3,y})$ comparing third order terms

Here we have an extra variable γ_1 which makes the solution not unique.

Reality condition

$\gamma_1 \in \mathbb{R}$ makes the solution unique.

Explicit (not very complicated:-) formulae for C_3 and γ_1

Solve $C_3 = (C_{3,x}, C_{3,y})$ comparing third order terms

Here we have an extra variable γ_1 which makes the solution not unique.

Reality condition

$\gamma_1 \in \mathbb{R}$ makes the solution unique.

Explicit (not very complicated:-) formulae for C_3 and γ_1

Homological equation

Solve $C_3 = (C_{3,x}, C_{3,y})$ comparing third order terms

Here we have an extra variable γ_1 which makes the solution not unique.

Reality condition

$\gamma_1 \in \mathbb{R}$ makes the solution unique.

Explicit (not very complicated:-) formulae for C_3 and γ_1

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.

In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

Example (Forced pendulum)

$$x''(t) = \sin(t) - \sin(x(t))$$

Goal: prove that there is an elliptic, stable periodic solution

Facts:

- This is is a Hamiltonian system
- $P(x) := \varphi(2\pi, x)$ is area preserving

Methodology:

- prove that there is a periodic orbit u_0 – Newton method
- bring $f(u) := P(u_0 + u) - u_0$ to the normal form
- check that $\gamma_1 \neq 0$ – twist condition

Moser's theorem guarantees that $u = 0$ is surrounded by f-invariant curves.
In consequence, periodic solution $t \to u_0(t)$ in the continuous system is stable and surrounded by 2D invariant tori.

```
/** Existence of invariant curves around ellipitic PO **/
#include <iostream>
#include "capd/capdlib.h"
#include "capd/normalForms/planarMaps.hpp"
using namespace capd;
int main(){
  IMap pendulum("time:t;var:x,dx;fun:dx,sin(t)-sin(x);",3);
  ICnOdeSolver solver(pendulum,20);
  ICnTimeMap tm(solver);
  // validate existence of a periodic point by Newton method
  IVector u0({0,-2.24910979679593});
  IVector r = interval(-1e-12,1e-12)*IVector{1,1};
  C0TripletonSet s0(u0);
  IVector y =  tm(2*interval::pi(),s0);
  C1Rect2Set s(u0+r);
  tm(2*interval::pi(),s);
  IMatrix DP = IMatrix(s) - IMatrix::Identity(2);
  IVector N = - matrixAlgorithms::gauss(DP,y-u0);
  std::cout << "subset(N,r)? = " << subset(N,r)
            << "\nN = " << N << "\nr = " << r << std::endl;
  //integrate 3rd order variational equations and compute normal form
  CnRect2Set S(u0+N,3);
  tm(2*interval::pi(),S);
  std::cout << "gamma1 = " <<
    normalForms::computePlanarEllipticNormalForm(S.currentSet())[1]
    << "\n(should have non-zero real part)";
  return 0;
}
```

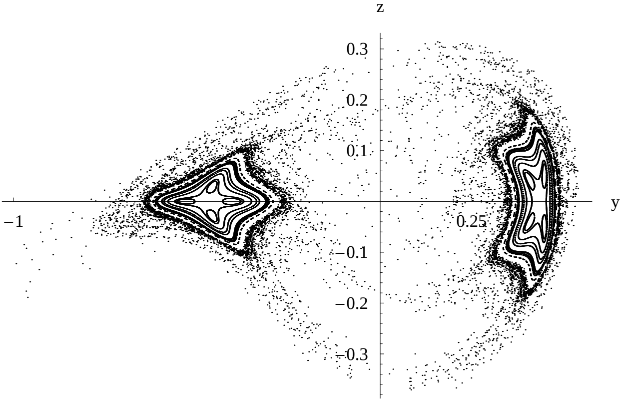
```
/* Output:
subset(N,r)? = 1
N = {[-4.465e-14, 4.22254e-14],[-1.45119e-14, 1.41125e-14]}
r = {[-1e-12, 1e-12],[-1e-12, 1e-12]}
gamma1 = ([0.937576, 0.937576],[-1.0637e-08, 1.0637e-08])
(should have non-zero real part)
*/
```

Remark:
The coefficients γ_i are always **real numbers**. The procedure, however, uses complex intervals.

Example (Michelson system)

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

Elliptic periodic orbits are observed for $c \in (0, 0.3194)$



Click picture to start animation

$$C_{elliptic} = (0.001, 0.3193749499054424068\,1)$$

Theorem

1. *There exists a continuous branch of symmetric, elliptic periodic orbits*

$$C_{elliptic} \ni c \to (0, y(c), 0) \in \mathbb{R}^3$$

2. *For*

$$c \in C_{stable} = C_{elliptic} \setminus (G_1 \cup G_2)$$

where

$$\begin{aligned} G_1 &= 0.2254404^{8933766649978}_{7596958760593} && \textit{(1:4 resonance)} \\ G_2 &= 0.27634^{347260295466508}_{298642570043203} && \textit{(1:3 resonance)} \end{aligned}$$

these orbits are stable:
each neighborhood of periodic orbit p_c contains 2D tori surrounding orbit p_c and invariant under the flow

$$C_{elliptic} = (0.001, 0.31937494990544240681)$$

Theorem

1. *There exists a continuous branch of symmetric, elliptic periodic orbits*

$$C_{elliptic} \ni c \to (0, y(c), 0) \in \mathbb{R}^3$$

2. *For*

$$c \in C_{stable} = C_{elliptic} \setminus (G_1 \cup G_2)$$

where

$$G_1 = 0.2254404^{8933766649978}_{7596958760593} \qquad \textit{(1:4 resonance)}$$
$$G_2 = 0.27634^{347260295466508}_{298642570043203} \qquad \textit{(1:3 resonance)}$$

these orbits are stable:
each neighborhood of periodic orbit p_c contains a 2D tori surrounding orbit p_c and invariant under the flow

$$C_{elliptic} = (0.001, 0.319374949905442406 81)$$

Theorem

1. *There exists a continuous branch of symmetric, elliptic periodic orbits*

$$C_{elliptic} \ni c \to (0, y(c), 0) \in \mathbb{R}^3$$

2. *For*

$$c \in C_{stable} = C_{elliptic} \setminus (G_1 \cup G_2)$$

where

$$G_1 = 0.2254404_{7596958760593}^{8933766649978} \quad \text{(1:4 resonance)}$$
$$G_2 = 0.27634_{298642570043203}^{347260295466508} \quad \text{(1:3 resonance)}$$

these orbits are stable:

each neighborhood of periodic orbit p_c contains a 2D tori surrounding orbit p_c and invariant under the flow.

$$C_{elliptic} = (0.001, 0.31937494990544240681)$$

Theorem

1. *There exists a continuous branch of symmetric, elliptic periodic orbits*

$$C_{elliptic} \ni c \to (0, y(c), 0) \in \mathbb{R}^3$$

2. *For*

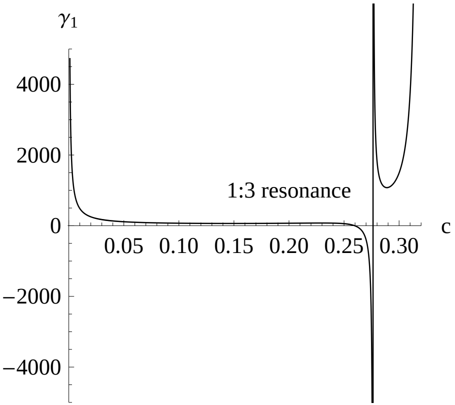
$$c \in C_{stable} = C_{elliptic} \setminus (G_1 \cup G_2)$$

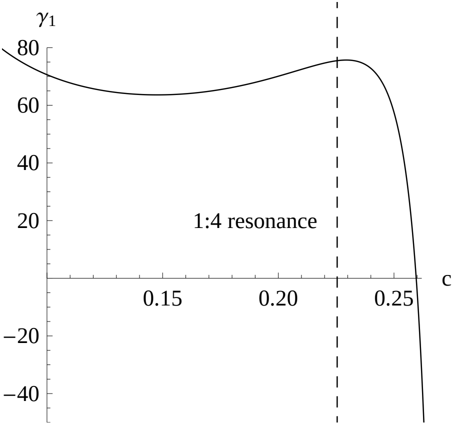
where

$$\begin{aligned} G_1 &= 0.2254404^{8933766649978}_{7596958760593} && \textit{(1:4 resonance)} \\ G_2 &= 0.27634^{347260295466508}_{298642570043203} && \textit{(1:3 resonance)} \end{aligned}$$

these orbits are stable:
each neighborhood of periodic orbit p_c contains a 2D tori surrounding orbit p_c and invariant under the flow.

Michelson system:





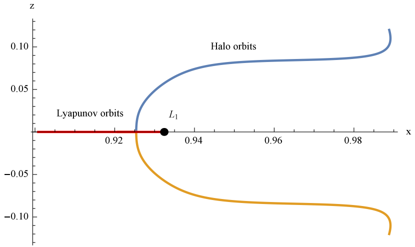
$$\mathrm{C}_{\text{stable}} \ni c \to \gamma_1(c) \in \mathbb{R}$$

Applications

Bifurcations of Halo orbits

We had theorem about the existence of a branch of Halo orbits

$$Z \ni z \to (x(z), 0, z, 0, \dot{y}(z), 0) \in \Pi$$
$$Z := [-0.083664781253492707, 0.083664781253492707]$$



Proved by validation of zeros of $f\colon \mathbb{R}^3 \to \mathbb{R}^2$ (for $z \geq z_0 > 0$)

$$f(z, x, \dot{y}) := \pi_{(\dot{x},\dot{z})} P(x, 0, z, 0, \dot{y}, 0)$$

Bifurcation for $z = 0$

Solution set is an intersection of two curves.
Interval Newton method must fail at intersection point.

Fact:

$$f_{\dot{z}}(z = 0, x, \dot{y}) \equiv 0$$

for all $(0, x, \dot{y})$ from the domain of $f_{\dot{z}}$.

Factorize by z:

$$f_{\dot{z}}(z, x, \dot{y}) = \int_0^1 \frac{d}{dt} f_{\dot{z}}(tz, x, \dot{y})dt = z \int_0^1 \frac{\partial}{\partial z} f_{\dot{z}}(tz, x, \dot{y})dt$$

Set

$$F(z, x, \dot{y}) = \int_0^1 \frac{\partial}{\partial z} f_{\dot{z}}(tz, x, \dot{y})dt$$

Now we have two equations:

$$\begin{aligned} \text{Lyapunov orbits} \quad &: \quad \{f_{\dot{x}} = 0 \wedge z = 0\} - \text{solved in 4D} \\ \text{Halo orbits} \quad &: \quad \{f_{\dot{x}} = 0 \wedge F = 0\} - \text{solved in 6D} \end{aligned}$$

Uniqueness

We expect that the solution sets to both equations are regular curves.

Lemma (Numerical method for bifurcation of Halo orbits)

$$
\begin{aligned}
W &= Z \times X \times \dot{Y} \\
&:= [-z_0, z_0] \times [x_0 - \Delta x, x_0 + \Delta x] \times [\dot{y}_0 - \Delta \dot{y}, \dot{y}_0 + \Delta \dot{y}] \\
N_1 &= \begin{bmatrix} x_0 \\ \dot{y}_0 \end{bmatrix} - \begin{bmatrix} \frac{\partial f_{\dot{x}}(W)}{\partial x} & \frac{\partial f_{\dot{x}}(W)}{\partial \dot{y}} \\ \frac{\partial^2 \mathbf{f}_{\dot{\mathbf{z}}}(\mathbf{W})}{\partial \mathbf{x} \partial \mathbf{z}} & \frac{\partial^2 \mathbf{f}_{\dot{\mathbf{z}}}(\mathbf{W})}{\partial \dot{\mathbf{y}} \partial \mathbf{z}} \end{bmatrix}_I^{-1} \cdot \begin{bmatrix} f_{\dot{x}}(Z, x_0, y_0) \\ \frac{\partial f_{\dot{x}}(Z, x_0, y_0)}{\partial z} \end{bmatrix} \\
N_2 &= x_0 - f_{\dot{x}}(0, x_0, \dot{Y}) \left[\frac{\partial f_{\dot{x}}}{\partial x}(0, X \times \dot{Y}) \right]^{-1}
\end{aligned}
$$

If $N_1 \subset \operatorname{int}(X \times \dot{Y})$ *and* $N_2 \subset \operatorname{int} X$ *then the solution set* $f(z, x, \dot{y}) = 0$ *restricted to* W *is the union of graphs of two smooth functions*

$$
\begin{aligned}
Z \ni z &\to (x(z), \dot{y}(z)) \in X \times \dot{Y} \\
\dot{Y} \ni \dot{y} &\to (0, x(\dot{y})) \in Z \times X.
\end{aligned}
$$

These curves intersect at exactly one point

$$(0, x(z = 0), \dot{y}(z = 0))$$