

The CAPD library: a tool for validation in dynamical systems

Daniel Wilczak

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

**International Workshop on Validated Numerics
for Computer-Assisted Proofs**

ICMS, Bayes Centre, Edinburgh, 09 July, 2026

CAPD library

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

<https://github.com/CAPDgroup>

1992–2026

continuously developed at the Jagiellonian University

CAPD library

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

<https://github.com/CAPDgroup>

1992–2026

continuously developed at the Jagiellonian University

- **CAPD::RedHom: computational topology**



Marian Mrozek



Paweł Pilarczyk



Mateusz Juda

- **CAPD::DynSys: computational dynamics**



Piotr Zgliczyński



Tomasz Kapela



Daniel Wilczak

The CAPD::DynSys in 2026

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

Some applications:

- $\mathcal{C}^0$ -computations;
chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$ -computations;
periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, connecting orbits
for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2 - \mathcal{C}^5$ computations;
global cocoon bifurcations, homoclinic tangencies, bifurcations of periodic orbits for ODEs, KAM stability of
elliptic periodic solutions, invariant tori
- PDEs;
chaos and connecting orbits

The CAPD::DynSys in 2026

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

Some applications:

- $\mathcal{C}^0$ -computations;
chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$ -computations;
periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, connecting orbits for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2 - \mathcal{C}^5$ computations;
global cocoon bifurcations, homoclinic tangencies, bifurcations of periodic orbits for ODEs, KAM stability of elliptic periodic solutions, invariant tori
- PDEs;
chaos and connecting orbits

The CAPD::DynSys in 2026

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

Some applications:

- $\mathcal{C}^0$ -computations;
chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$ -computations;
periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, connecting orbits for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2 - \mathcal{C}^5$ computations;
global cocoon bifurcations, homoclinic tangencies, bifurcations of periodic orbits for ODEs, KAM stability of elliptic periodic solutions, invariant tori
- PDEs;
chaos and connecting orbits

The CAPD::DynSys in 2026

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

Some applications:

- $\mathcal{C}^0$ -computations;
chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$ -computations;
periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, connecting orbits for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2 - \mathcal{C}^5$ computations;
global cocoon bifurcations, homoclinic tangencies, bifurcations of periodic orbits for ODEs, KAM stability of elliptic periodic solutions, invariant tori
- PDEs;
chaos and connecting orbits

The CAPD::DynSys in 2026

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

Some applications:

- $\mathcal{C}^0$ -computations;
chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$ -computations;
periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, connecting orbits for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2 - \mathcal{C}^5$ computations;
global cocoon bifurcations, homoclinic tangencies, bifurcations of periodic orbits for ODEs, KAM stability of elliptic periodic solutions, invariant tori
- PDEs;
chaos and connecting orbits

The CAPD::DynSys in 2026

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

Some applications:

- $\mathcal{C}^0$ -computations;
chaotic dynamics for many textbook systems, bifurcations for reversible systems
- $\mathcal{C}^1$ -computations;
periodic orbits (in quite high dimensions, like 300 for the N-body problem), hyperbolicity, connecting orbits for ODEs both to equilibria and periodic solutions
- $\mathcal{C}^2 - \mathcal{C}^5$ computations;
global cocoon bifurcations, homoclinic tangencies, bifurcations of periodic orbits for ODEs, KAM stability of elliptic periodic solutions, invariant tori
- PDEs;
chaos and connecting orbits

Main header files

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

Defined types:

- interval, MpFloat, MpInterval
- **Algebra:**
 - DVector, LDVector, IVector, MpVector, MplVector
 - [Prefix]Matrix
 - [Prefix]Hessian
 - [Prefix]Jet
- **Automatic differentiation:**
 - [Prefix]Map
- **ODE solvers:**
 - [Prefix]OdeSolver, [Prefix]CnOdeSolver
 - [Prefix]TimeMap, [Prefix]CnTimeMap
 - [Prefix]PoincareMap, [Prefix]CnPoincareMap

Main header files

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

Defined types:

- interval, MpFloat, MpInterval
- Algebra:
 - DVector, LDVector, IVector, MpVector, MplVector
 - [Prefix]Matrix
 - [Prefix]Hessian
 - [Prefix]Jet
- Automatic differentiation:
 - [Prefix]Map
- ODE solvers:
 - [Prefix]OdeSolver, [Prefix]CnOdeSolver
 - [Prefix]TimeMap, [Prefix]CnTimeMap
 - [Prefix]PoincareMap, [Prefix]CnPoincareMap

Main header files

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

Defined types:

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

Main header files

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

Defined types:

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

Main header files

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

Defined types:

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

Main header files

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

Defined types:

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

```

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

int main(){
    IMap f("par:a;var:x,y;fun:sin(x*cos(y)),y*cos(a*y)-x^2;",3);
    f.setParameter("a",interval(1.,1.01));

    IVector u({1,3});
    cout << "f(u)=" << f(u) << endl;
    cout << "Df(u)=" << f.derivative(u) << endl;

    IJet jet(2,3); // third order Taylor coefficients, two variables
    f(u,jet);      // compute full Taylor expansion
    cout << jet.toString() << endl;
}

```

```

/** 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
    ClHOREct2Set 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.);
    ClHOREct2Set 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;
}

```

```
/* 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;
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);

    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);
    std::cout << "P(1,5,27)=" << P << std::endl;
    std::cout << "DP(1,5,27)=" << DP << std::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, 2.5335e-13] }
}
*/

```


Examples & Applications

- symmetric periodic orbits ($\mathcal{C}^0$ computation)

D. Michelson,

Steady solutions of the Kuramoto-Sivashinsky equation, Physica D: Nonlinear Phenomena 19 (1) (1986) 89–111.

- BVP ($\mathcal{C}^1$ computation)

M. T. Nakao,

A numerical verification method for the existence of weak solutions for nonlinear boundary value problems, Journal of Mathematical Analysis and Applications 164 (2) (1992) 489 – 507.

- optimization ($\mathcal{C}^2$ computation)

- existence of attractor, symbolic dynamics and bifurcations

O. E. Rössler,

An equation for continuous chaos, Phys. Lett. A 57 (5) (1976) 397–398.

P. Zgliczyński,

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

D. Wilczak, S. Serrano and R. Barrio

A mechanism for growth of topological entropy. Chaos, 35:103117, 2025.

- existence of hyperbolic attractor

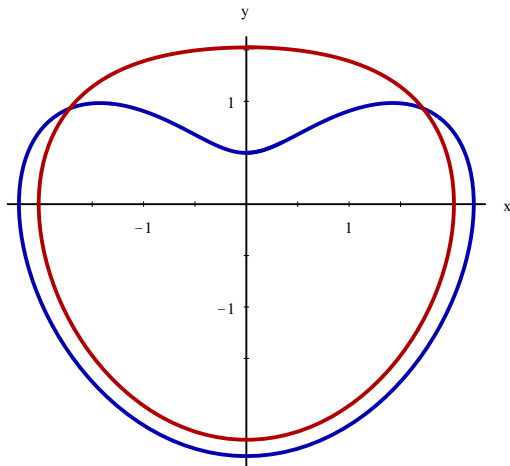
D. Wilczak,

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

Example (D. Michelson, Physica D, 1986)

$$x''' = 1 - x' - x^2/2$$

Prove the existence of two odd periodic solutions.



Setting:

$$\begin{aligned}\Pi &= \{(0, y, z) : y, z \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi &\rightarrow \Pi & - & \text{Poincaré map}\end{aligned}$$

Reversing symmetry:

$$R(y, z, t) = (y, -z, -t)$$

Shooting method:

- If $\pi_z P(y_*, 0) = 0$ then $(y_*, 0)$ is a symmetric periodic point of P
- Search for zeroes of $F(y) = \pi_z P(y, 0)$
- Check if F is continuous on $[y_1, y_2]$ and

$$F(y_1) \cdot F(y_2) < 0$$

Setting:

$$\begin{array}{ll}\Pi = \{(0, y, z) : y, z \in \mathbb{R}\} & - \text{Poincaré section} \\ P : \Pi \rightarrow \Pi & - \text{Poincaré map}\end{array}$$

Reversing symmetry:

$$R(y, z, t) = (y, -z, -t)$$

Shooting method:

- If $\pi_z P(y_*, 0) = 0$ then $(y_*, 0)$ is a symmetric periodic point of P
- Search for zeroes of $F(y) = \pi_z P(y, 0)$
- Check if F is continuous on $[y_1, y_2]$ and

$$F(y_1) \cdot F(y_2) < 0$$

Setting:

$$\begin{aligned}\Pi &= \{(0, y, z) : y, z \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi &\rightarrow \Pi & - & \text{Poincaré map}\end{aligned}$$

Reversing symmetry:

$$R(y, z, t) = (y, -z, -t)$$

Shooting method:

- If $\pi_z P(y_*, 0) = 0$ then $(y_*, 0)$ is a symmetric periodic point of P
- Search for zeroes of $F(y) = \pi_z P(y, 0)$
- Check if F is continuous on $[y_1, y_2]$ and

$$F(y_1) \cdot F(y_2) < 0$$

Setting:

$$\begin{aligned}\Pi &= \{(0, y, z) : y, z \in \mathbb{R}\} & - & \text{Poincaré section} \\ P : \Pi &\rightarrow \Pi & - & \text{Poincaré map}\end{aligned}$$

Reversing symmetry:

$$R(y, z, t) = (y, -z, -t)$$

Shooting method:

- If $\pi_z P(y_*, 0) = 0$ then $(y_*, 0)$ is a symmetric periodic point of P
- Search for zeroes of $F(y) = \pi_z P(y, 0)$
- Check if F is continuous on $[y_1, y_2]$ and

$$F(y_1) \cdot F(y_2) < 0$$

```

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

void validateSymPO(MpInterval y){
    MpIMap vf("var:x,y,z;fun:y,z,1-y-0.5*x^2;"); // vector field
    MpIOdeSolver solver(vf, 80); // 80th order solver
    MpICoordinateSection section(3, 0); // section x=0
    MpIPoincareMap pm(solver, section);

    // Check that the Poincare map is defined on (y,0)
    pm({MpInterval(0.),y,MpInterval(0.)});
    MpIVector L = pm({MpInterval(0.),y.left(),MpInterval(0.)});
    MpIVector R = pm({MpInterval(0.),y.right(),MpInterval(0.)});
    std::cout << "L=" << L[2] << ", R=" << R[2] << std::endl;
    std::cout << "inequality? " << ( L[2]*R[2]<0 ) << std::endl;
}

int main(){
    std::cout.precision(51);
    MpFloat::setDefaultPrecision(200);
    MpInterval I = MpInterval(-1,1)*1e-50;
    validateSymPO("1.52596173050368837661290525530252178585479114593993" + I);
    validateSymPO("0.500025648535208099091780440333115543919649610703305" + I);
}

```

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:

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

Neumann BVP:

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

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:

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

Neumann BVP:

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

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:

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

Neumann BVP:

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

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:

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

Neumann BVP:

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

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:

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

Neumann BVP:

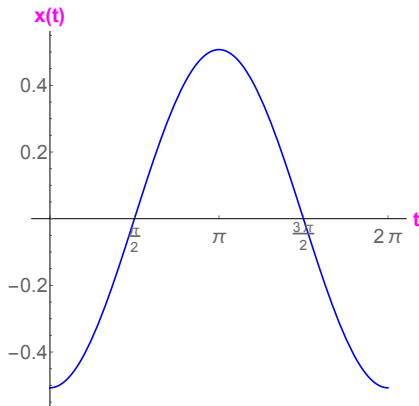
$$x'(0) = A, \quad x'(T) = B \quad \rightsquigarrow \quad F(x) = \pi_{x'}(\varphi(T, (x, A))) - B$$

Solve by interval Newton (Krawczyk) method

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

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

Prove the existence of solution satisfying $x'(0) = x'(2\pi) = 0$.



Method:

solve

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

using interval Newton method

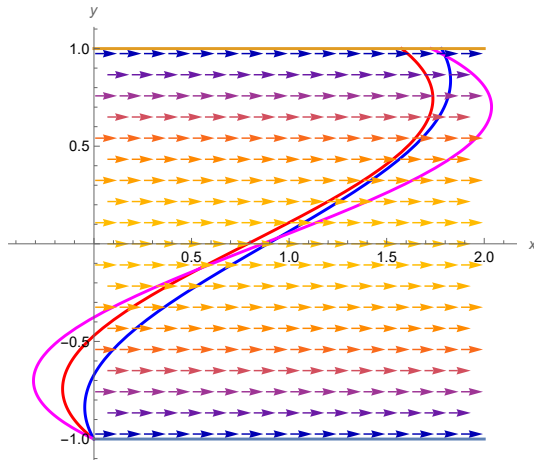
```

/** Example of solving BVP:  $x'(0)=x'(2\pi)=0$  */
#include <iostream>
#include "capd/capdlib.h"
using namespace capd;
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-4,1e-4),0.});
    COHOTripletonSet s0(u0);
    ClHOREct2Set s(u0+r);
    // integrate until  $T=2\pi$ 
    IVector y = tm(2.*interval::pi(),s0);
    tm(2.*interval::pi(),s);
    // solve equation  $F(r_1) := \text{proj}_{\{x'\}}(\phi(2\pi,u_0+(r_1,0))) = 0$ 
    interval N = - y[1]/((IMatrix)s)(2,1);
    std::cout << "(N,r1)=(" << N << ", " << r[0] << ")"<< std::endl;
    std::cout << "subset(N,r)? = " << std::boolalpha << subset(N,r[0]) << std::endl;
}
/* Output:
(N,r1)=([-3.84493e-05, -2.09823e-05],[-0.1, 0.1])
subset(N,r)? = true
*/

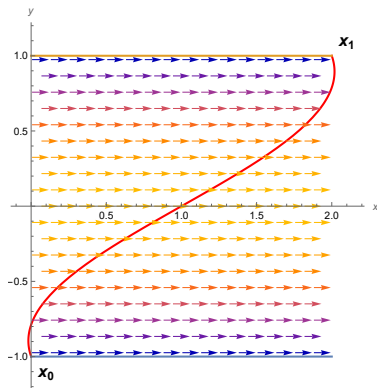
```

Optimization problem



River:

- river of constant width $y \in [-L, L]$
- current of the river $V(x, y) = (V_1(x, y), 0)$

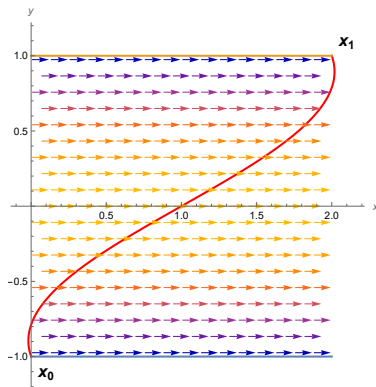


Swimmer:

- constant velocity and angle $\vec{v} = v(\cos \alpha, \sin \alpha)$, $v \ll V_1$
- in y direction we have $y(t) = -L + tv \sin \alpha$
- swim time: $T(\alpha) = 2L/(v \sin \alpha)$.

River:

- river of constant width $y \in [-L, L]$
- current of the river $V(x, y) = (V_1(x, y), 0)$

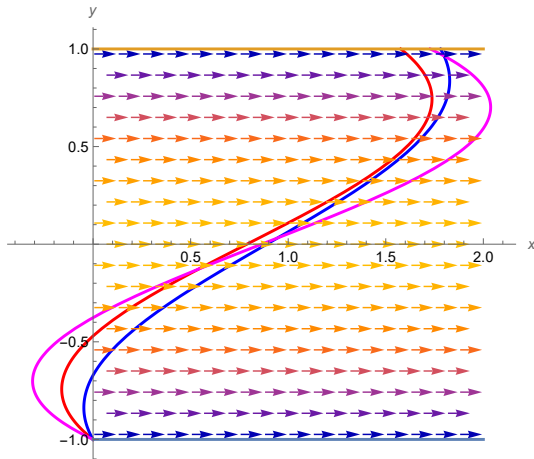


Swimmer:

- constant velocity and angle $\vec{v} = v(\cos \alpha, \sin \alpha)$, $v \ll V_1$
- in y direction we have $y(t) = -L + tv \sin \alpha$
- swim time: $T(\alpha) = 2L/(v \sin \alpha)$.

Nonautonomous ODE:

$$\begin{aligned}x'(t) &= V_1(x(t), y(t)) + v \cos \alpha \\ &= V_1(x(t), tv \sin \alpha - L) + v \cos \alpha\end{aligned}$$



Problem

Find an angle so that the drift will be minimal.

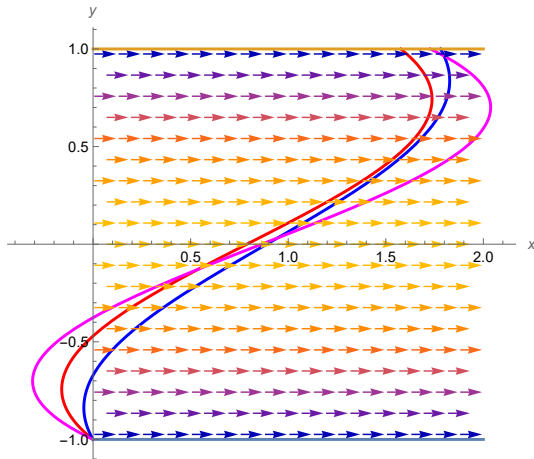
Method

Find global minimum of:

$$g(\alpha) := \phi(T(\alpha), x_0) = x_0(2L/(v \sin \alpha))$$

Nonautonomous ODE:

$$\begin{aligned}x'(t) &= V_1(x(t), y(t)) + v \cos \alpha \\ &= V_1(x(t), tv \sin \alpha - L) + v \cos \alpha\end{aligned}$$



Problem

Find an angle so that the drift will be minimal.

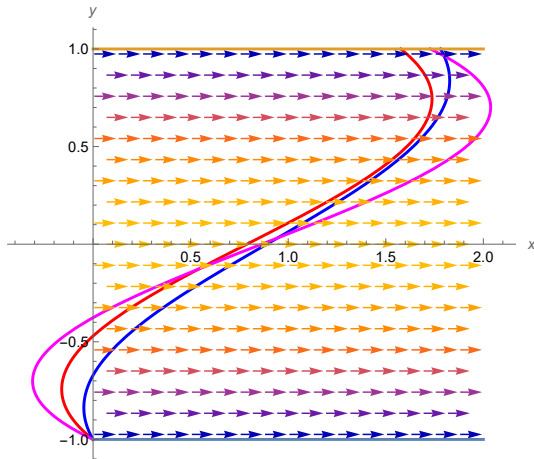
Method

Find global minimum of:

$$g(\alpha) := \phi(T(\alpha), x_0) = x_0(2L/(v \sin \alpha))$$

Nonautonomous ODE:

$$\begin{aligned}x'(t) &= V_1(x(t), y(t)) + v \cos \alpha \\ &= V_1(x(t), tv \sin \alpha - L) + v \cos \alpha\end{aligned}$$



Problem

Find an angle so that the drift will be minimal.

Method

Find global minimum of:

$$g(\alpha) := \phi(T(\alpha), x_0) = x_0(2L/(v \sin \alpha))$$

Data:

- $L = 1$
- $\nu = 0.5$
- $V_1(x, y) = \frac{1}{8}(\sin(x) + 1) + \cos(\frac{1}{2}\pi y)$
- $x_0 = 0$

Equation to solve:

$$g'(\alpha) = f(T(\alpha), x_0)T'(\alpha) + \frac{\partial \phi}{\partial \alpha}(T(\alpha), x_0) = 0$$

```

#include "capd/capdlib.h"
using namespace capd;

int main() {
    IMap vf("par:pi;time:t;var:x,a;fun: (sin(x)+1)/8+cos(pi*(t*sin(a)/2-1)/2)+cos(a)/2,0;",2);
    vf.setParameter("pi",interval::pi());
    IC2OdeSolver solver(vf,6);
    IC2TimeMap tm(solver);

    // Approximate parameter:
    interval a = 2.242655603792935;

    interval T = 4.0/sin(a);
    C1HOREct2Set s1(IVector{0.,a});
    IVector z = tm(T,s1);
    // f(T(a),a)*T'(a) + x_a(T(a),a)
    interval dg = -4.0*vf(T,z)[0]*cos(a)/sqr(sin(a)) + ((IMatrix)s1)[0][1];

    // travelling time and its derivatives wrt angle
    double e = 1e-12;
    interval A = a+interval(-e,e), s = sin(A), c = cos(A);
    T = 4.0/s;
    interval dT = -4.0*c/sqr(s);
    interval d2T = 4.0*(sqr(c)+1)/power(s,3);

    C2Rect2Set s2(IVector{0.,A});
    z = tm(T,s2);
    IVector coeffs[3]={z,z,z}; // just to init with something
    IMatrix D = vf.derivative(T,z);
    vf.computeODECoefficients(coeffs,2);
    // (x_tt*T' + 2vf_a)*T' + x_t T'' + x_aa
    interval d2g = 2.0*dT*(coeffs[2][0]*dT+D[0][1]) + coeffs[1][0]*d2T + 2.*s2(0,1,1);
    interval N = -dg/d2g;
    std::cout << "N=" << N << "\nResult: " << (abs(N)<e) << std::endl;
}

```

Roberto Barrio



Sergio Serrano

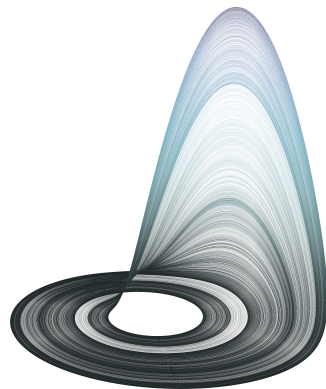
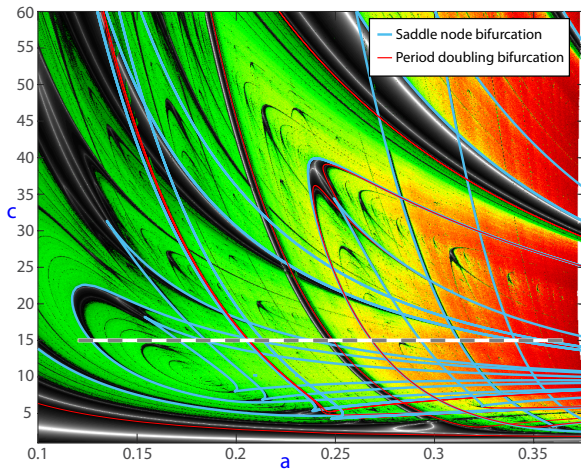


A mechanism for growth of topological entropy, Chaos vol. 35 (2025), 103117.

Example (Rössler system)

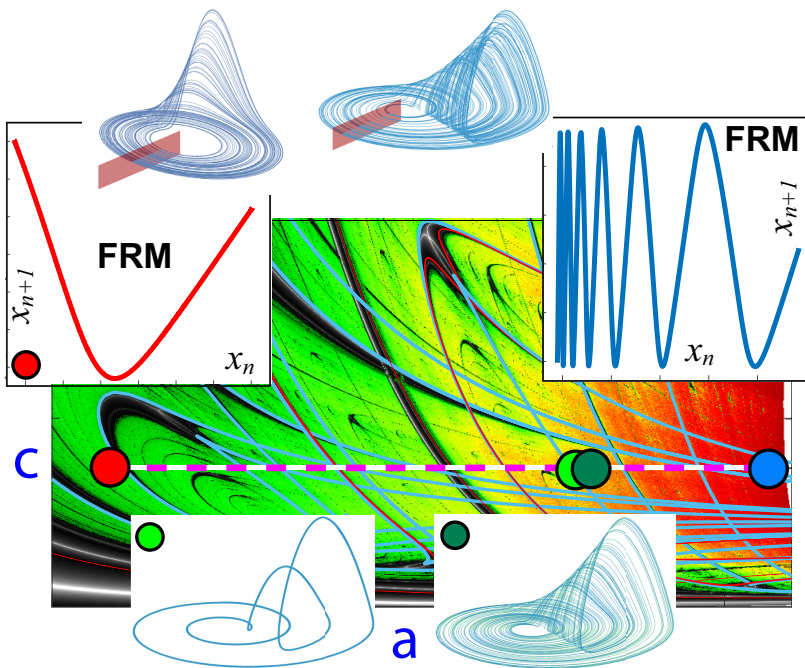
$$x' = -(y + z), \quad y' = x + ay, \quad z' = b + z(x - c)$$

O.E. Rössler. An equation for continuous chaos, Physics Letters A 1976.



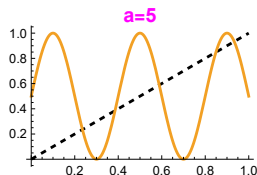
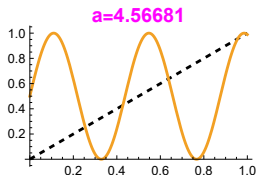
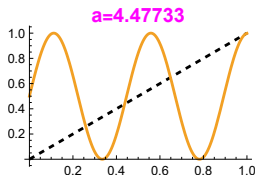
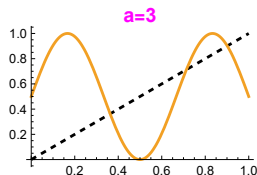
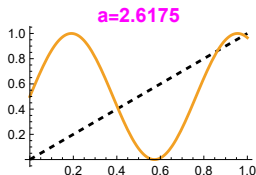
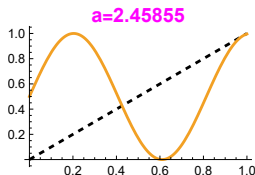
$$a = b = 0.2, c = 5.7$$

DW, S. Serrano, R. Barrio. A mechanism for growth of topological entropy, **Chaos** 2025.



TOY MODEL: the sine map

$$f_a(x) = \frac{1}{2}(1 + \sin(a\pi x)), \quad a \in \mathbb{R}, \quad x \in [0, 1].$$



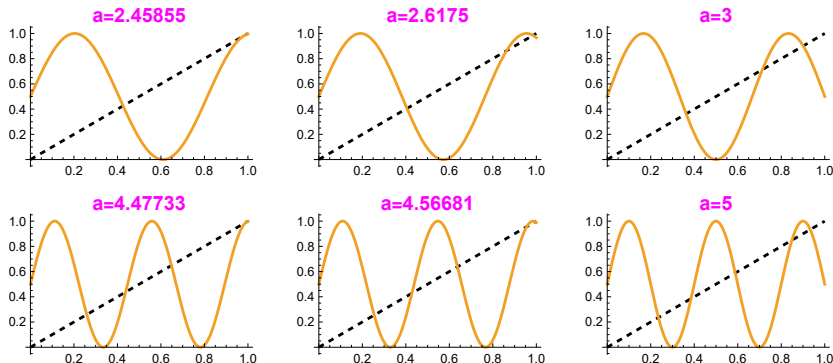
Left: fold bifurcation creates a stable/unstable fixed-point pair.

Middle: stable point loses stability (period-doubling).

Right: new monotone branch $\Rightarrow$ new symbol.

TOY MODEL: the sine map

$$f_a(x) = \frac{1}{2}(1 + \sin(a\pi x)), \quad a \in \mathbb{R}, \quad x \in [0, 1].$$

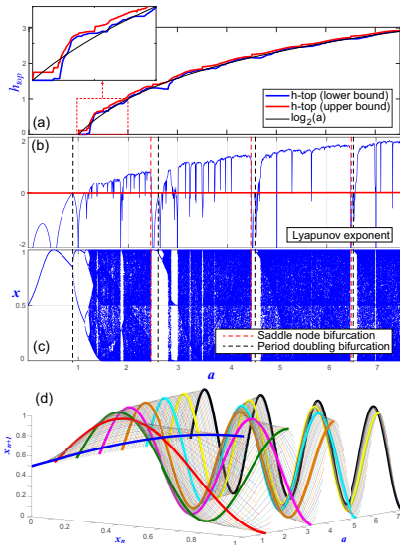


Left: fold bifurcation creates a stable/unstable fixed-point pair.

Middle: stable point loses stability (period-doubling).

Right: new monotone branch $\Rightarrow$ new symbol.

TOY MODEL: the sine map $f_a(x) = \frac{1}{2}(1 + \sin(a\pi x))$, $a \in \mathbb{R}$.



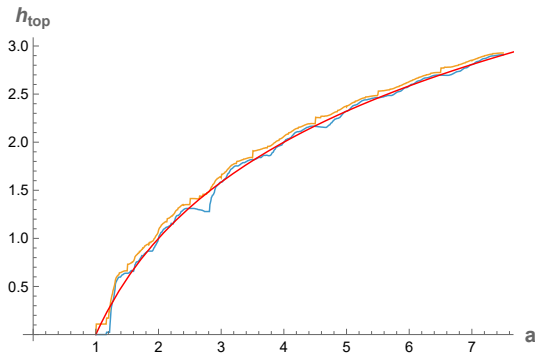
(a): Lower & upper bounds on h_{top} for $a \in [0, 7.5]$. Entropy grows like $\log_2 a$.

(b): Lyapunov exponent.

(c): Bifurcation diagram — sequence of saddle-node bifurcations, each followed by period-doubling cascade.

(d): Graph of f_a — number of monotone branches increases with a .

TOY MODEL: the sine map $f_a(x) = \frac{1}{2}(1 + \sin(a\pi x))$, $a \in \mathbb{R}$.



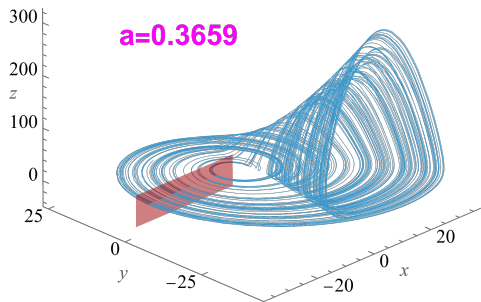
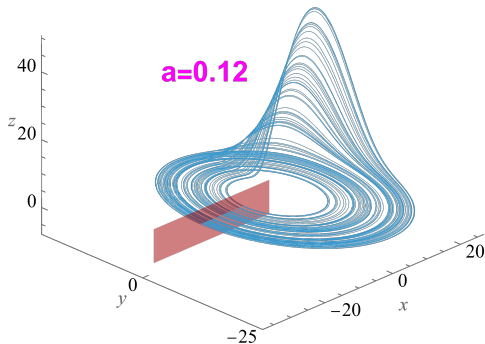
Algorithms

- $h_{\text{top}}(f) = \frac{1}{k} h_{\text{top}}(f^k)$
- **lower bound:** semi-conjugacy to shift of finite type
- **upper bound:** logarithm of the number of monotone slopes

Poincaré map in the Rössler system

Rössler system

$$x' = -(y + z), \quad y' = x + ay, \quad z' = b + z(x - c)$$



Poincaré section

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

Theorem 1 — Existence of a trapping region

Theorem 1

For all $a \in \mathcal{A} = [a_{\min}, a_{\max}] = [0.12, 0.3659]$, the Poincaré map $P_a : \Pi \rightarrow \Pi$ is **well-defined and smooth** on

$$\mathcal{T} = [-30.53, -3] \times [0.004, 0.011] \subset \Pi$$

and

$$P_a(\mathcal{T}) \subset \text{int } \mathcal{T}.$$

Consequences

- $\mathcal{T}$ is a **trapping region**: trajectories never leave once inside.
- Maximal invariant set

$$\mathcal{I}_a = \bigcap_{n \geq 0} P_a^n(\mathcal{T})$$

is non-empty, compact, and connected.

- $\mathcal{I}_a$ contains **an attractor** of P_a .

Theorem 1 — Existence of a trapping region

Theorem 1

For all $a \in \mathcal{A} = [a_{\min}, a_{\max}] = [0.12, 0.3659]$, the Poincaré map $P_a : \Pi \rightarrow \Pi$ is **well-defined and smooth** on

$$\mathcal{T} = [-30.53, -3] \times [0.004, 0.011] \subset \Pi$$

and

$$P_a(\mathcal{T}) \subset \text{int } \mathcal{T}.$$

Consequences

- $\mathcal{T}$ is a **trapping region**: trajectories never leave once inside.
- Maximal invariant set

$$\mathcal{I}_a = \bigcap_{n>0} P_a^n(\mathcal{T})$$

is non-empty, compact, and connected.

- $\mathcal{I}_a$ contains **an attractor** of P_a .

Proof (CAP)

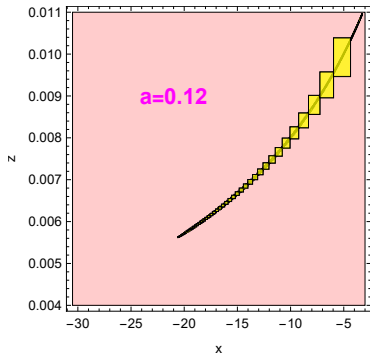
Step 1. Existence of $P_a(u)$ for all $(a, u) \in \mathcal{A} \times \mathcal{T}$:

→ 36 316 641 adaptive boxes.

Step 2. $P_a(\partial\mathcal{T}) \subset \text{int } \mathcal{T}$:

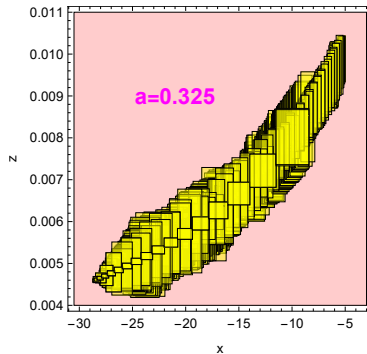
→ 9 299 383 adaptive boxes.

Jordan theorem $\Rightarrow P_a(\mathcal{T}) \subset \text{int } \mathcal{T}$.



Pink:
trapping region $\mathcal{T}$

Yellow:
bound of $P_a(\partial\mathcal{T})$



Relevant extrema

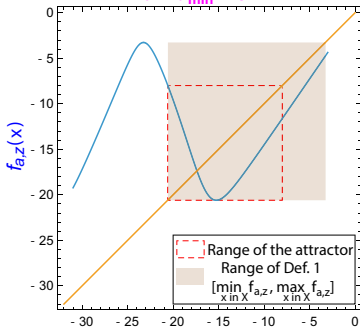
For fixed a and $z \in [z_{\min}, z_{\max}]$, define $f_{a,z}(x) := \pi_x P_a(x, z)$.

Definition — relevant extremum

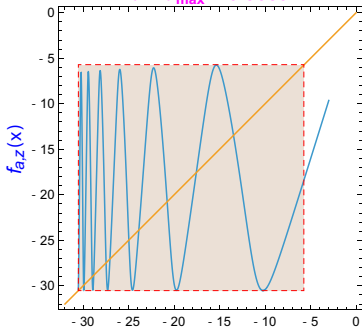
$x_* \in \text{int } X$ is a *relevant extremum* of f if it is a strict local extremum and $\min_X f \leq x_* \leq \max_X f$.

$\text{relEx}(f)$ = number of relevant extrema.

$a = a_{\min} = 0.12$



$a = a_{\max} = 0.3659$

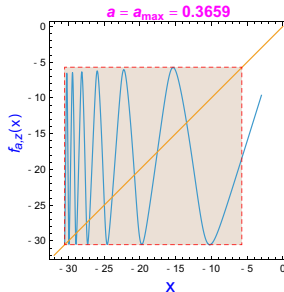
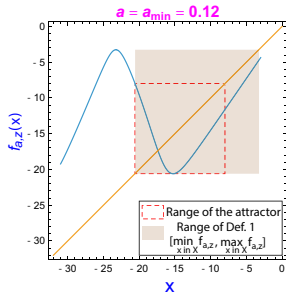


- Extrema *outside* the range of f do not affect the invariant-set structure.
- relEx is an **invariant of topological conjugacy** of interval maps.
- Changes in relEx signal **global topological changes** in $\mathcal{I}_a$.

Theorem 2

For the parameter values listed below, $\text{relEx}(f_{a,z})$ is independent of $z \in [z_{\min}, z_{\max}]$ and equals:

a	$a_{\min}$	.20	.26	.30	.32	.333	.345	.35	.356	.36	.362	$a_{\max}$
relEx	1	2	3	4	5	6	7	8	9	10	11	12



Why this matters

- Between each pair of consecutive parameter values, a **global topological change** of $\mathcal{I}_a$ occurs.
- relEx counts the number of “folds” in $P_a(\mathcal{T})$.
- Each new fold is preceded by a new saddle-node bifurcation.

Proof (CAP)

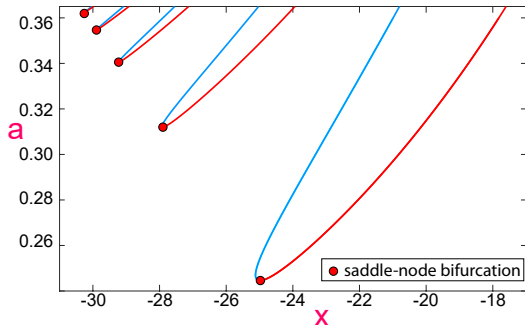
Step 1. Find subintervals in $[x_{\min}, x_{\max}]$ where $f_{a,z}$ is monotone

Step 2. Check additionally convexity if $f'_{a,z}$ is possibly zero.

x	$f_{a,z}(x)$	$f'_{a,z}(x)$	$f''_{a,z}(x)$
$[-30.53000000000000, -30.234918150004624]$	$[-30.19, -6.57]$	$[0.001, 184]$	—
$[-30.234918150004624, -30.234487860323142]$	$[-6.58, -6.57]$	$[-0.72, 0.58]$	$[-1326, -325]$
$[-30.234487860323142, -29.92685079785111]$	$[-30.44, -6.57]$	$[-230, -0.001]$	—
$[-29.92685079785111, -29.925276370697883]$	$[-30.44, -30.43]$	$[-0.89, 2.63]$	$[101, 1193]$
$[-29.925276370697883, -29.433781614317695]$	$[-30.44, -6.47]$	$[0.0007, 122]$	—
$[-29.433781614317695, -29.432550550415712]$	$[-6.48, -6.47]$	$[-0.77, 0.59]$	$[-578, -68]$
$[-29.432550550415712, -28.936271584068677]$	$[-30.45, -6.47]$	$[-138, -0.009]$	—
$[-28.936271584068677, -28.933364707472052]$	$[-30.45, -30.44]$	$[-0.87, 1.87]$	$[73, 418]$
$[-28.933364707472052, -28.134274742819773]$	$[-30.45, -6.36]$	$[0.006, 75.5]$	—
$[-28.134274742819773, -28.129878233661032]$	$[-6.37, -6.36]$	$[-1.2, 0.66]$	$[-227, -18]$
$[-28.129878233661032, -27.315443495151083]$	$[-30.47, -6.36]$	$[-86.7, -0.04]$	—
$[-27.315443495151083, -27.309055771711787]$	$[-30.47, -30.45]$	$[-1.66, 1.72]$	$[23, 156]$
$[-27.309055771711787, -25.978724324556815]$	$[-30.49, -6.23]$	$[0.004, 31.4]$	—
$[-25.978724324556815, -25.969275484986035]$	$[-6.24, -6.23]$	$[-0.83, 0.73]$	$[-79, -9.6]$
$[-25.969275484986035, -24.583975022884346]$	$[-30.49, -6.23]$	$[-49.91, -0.01]$	—
$[-24.583975022884346, -24.573109921220805]$	$[-30.5, -30.45]$	$[-1.1, 1.2]$	$[7.2, 56]$
$[-24.573109921220805, -22.263096727912796]$	$[-30.48, -6.05]$	$[0.01, 17.7]$	—
$[-22.263096727912796, -22.244978977377045]$	$[-6.06, -6.05]$	$[-0.68, 0.71]$	$[-29.2, -0.2]$
$[-22.244978977377045, -19.75178920778561]$	$[-30.5, -6.05]$	$[-21.8, -0.02]$	—
$[-19.75178920778561, -19.732234778220185]$	$[-30.5, -30.49]$	$[-0.36, 0.36]$	$[0.15, 19]$
$[-19.732234778220185, -15.379434543948365]$	$[-30.5, -5.74]$	$[0.0007, 8.1]$	—
$[-15.379434543948365, -15.345294934267804]$	$[-5.75, -5.73]$	$[-0.51, 0.51]$	$[-7.9, -0.2]$
$[-15.345294934267804, -10.256384670433105]$	$[-30.53, -5.74]$	$[-7.8, -0.004]$	—
$[-10.256384670433105, -10.216471648755972]$	$[-30.54, -30.52]$	$[-0.2, 0.2]$	$[1.6, 2.6]$
$[-10.216471648755972, -3]$	$[-30.54, -9.63]$	$[0.01, 3.3]$	—

Theorem 3

- P_a undergoes at least five saddle-node bifurcation in the considered range of parameters.
- The non-bifurcation eigenvalue of DP_a at bifurcation point satisfies $|\lambda_i| < 2 \cdot 10^{-4}$.
- In each case the two branches of fixed points of P_a , born at a_i^* , exist for $a \in [a_i^*, a_{\max}]$.



Stable and **unstable** branches of fixed points of P_a .

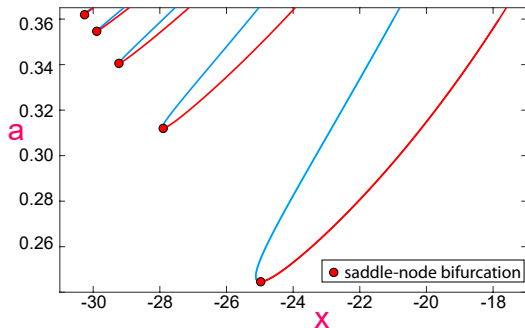
Above each a_i^* : an **attracting periodic orbit** is created, coexisting with the chaotic invariant set.

Proof (CAP)

Step 1. Bifurcation: parameterize by $x \rightarrow (z(x), a(x))$ and prove local minimum of $a(x)$

Step 2. Continuation: parameterize by $a \rightarrow (x(a), z(a))$.

i	x_i	z_i	a_i
1	-24.98615641824929	0.005003695953767296	0.2445890212249042
2	-27.90101006311546	0.004663547021688063	0.3119866509093180
3	-29.22211573599117	0.004524155280447693	0.3405236989996505
4	-29.89377365336397	0.004456435009829962	0.3546641965836549
5	-30.25817997684925	0.004420535075303823	0.3623180183723328



Bounds on bifurcation points

$$|x_i^* - x_i| \leq 3 \cdot 10^{-11}$$

$$|z_i^* - z_i| \leq 10^{-14}$$

$$|a_i^* - a_i| \leq 6 \cdot 10^{-14}$$

Tool for topological horseshoe

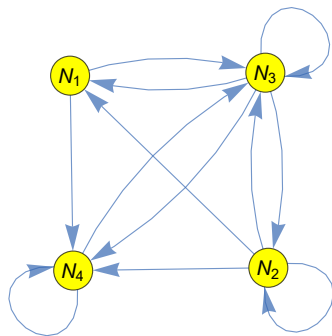
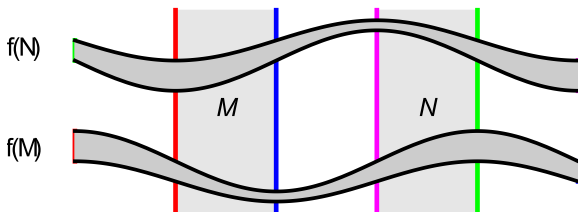
Theorem (Zgliczyński '1997)

Assume $\{N_i\}_{i=1}^k$ are pairwise disjoint rectangles and set

$$A_{ij} = \begin{cases} 1 & \text{if } N_i \xrightarrow{f} N_j \\ 0 & \text{otherwise} \end{cases}$$

Then

- there is an invariant set $\mathcal{I}$ such that $f|_{\mathcal{I}}$ is semi-conjugate to a subshift of finite type with transition matrix A .
- every periodic sequence of symbols leads to periodic point for f with the same principal period.



Tool for topological horseshoe

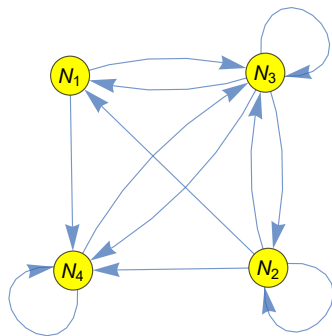
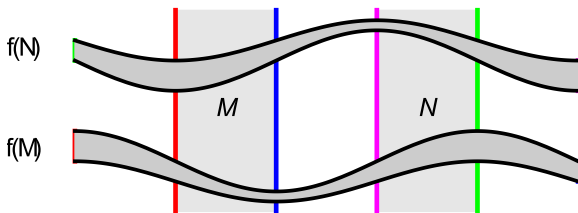
Theorem (Zgliczyński '1997)

Assume $\{N_i\}_{i=1}^k$ are pairwise disjoint rectangles and set

$$A_{ij} = \begin{cases} 1 & \text{if } N_i \xrightarrow{f} N_j \\ 0 & \text{otherwise} \end{cases}$$

Then

- there is an invariant set $\mathcal{I}$ such that $f|_{\mathcal{I}}$ is semi-conjugate to a subshift of finite type with transition matrix A .
- every periodic sequence of symbols leads to periodic point for f with the same principal period.



Theorem 4 — Growth of topological entropy

- For all $a \in [a_{\min}, a_{\max}]$, there exists an invariant set $\mathcal{H}_a \subset \mathcal{T}$ such that $P_a|_{\mathcal{H}_a}$ is semi-conjugate to a subshift of finite type with positive topological entropy.
- The number of symbols grows from **2** (at $a = a_{\min}$) to **13** (at $a = a_{\max}$).

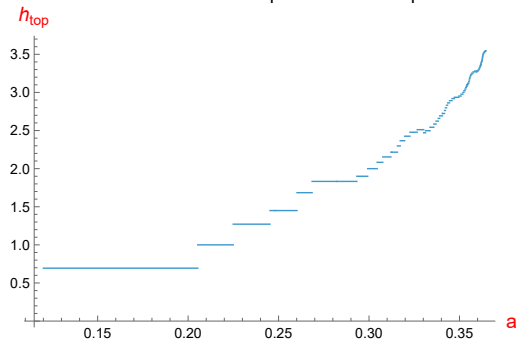
Leading transition matrices as the number of symbols grows:

$$M_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, M_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, M_4 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, M_5 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \dots$$

For $a = a_{\min}$: $h_{\text{top}}(M_1) = \log_2 \left(\frac{1}{2} (1 + \sqrt{5}) \right) \approx 0.694$.

For $a = a_{\max}$: $h_{\text{top}}(M_{86}) = \log_2 12 \approx 3.585$:

$$M_{86} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

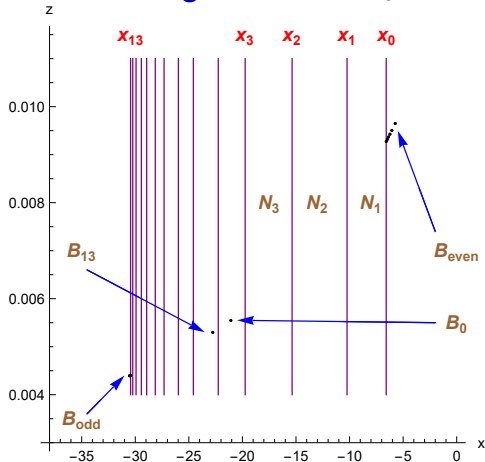


Proof (CAP)

Step 1. Detect approximate extrema of FRM and define rectangles

Step 2. Compute image of edges and check semi-conjugacy to shift dynamics.

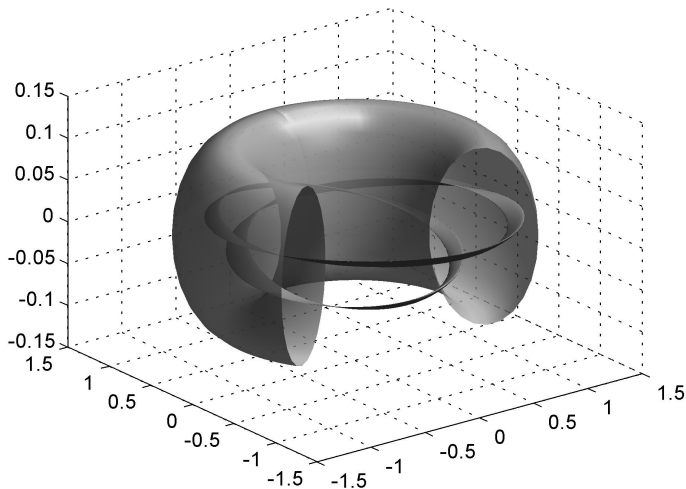
Rectangles for $a = a_{\max}$



Computation

- adaptive subdivision of domain and parameter
- **over 2 hours on 144 threads**

Hyperbolic attractor



DW, *Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system*, SIADS'2010.

Some theoretical tools for proving hyperbolicity

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.

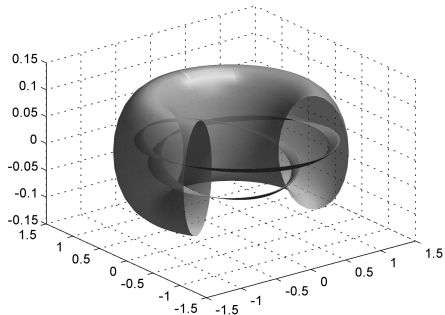
DW, Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system, *SIADS 2010*, Vol. 9, 1263–1283.

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

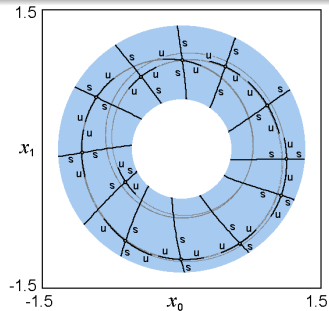
Example (Kuznetsov system)

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

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



$$\begin{aligned} \omega_0 &= 2\pi \\ A &= 5 \\ T &= 6 \\ \varepsilon &= 0.5 \end{aligned}$$



Theorem

Put $P(x) = \varphi(T, x)$, where $T = 6$ – period of vector field

There exists compact, connected, explicit set $\mathcal{B}$ ($\sim 8 \cdot 10^6$ boxes) such that

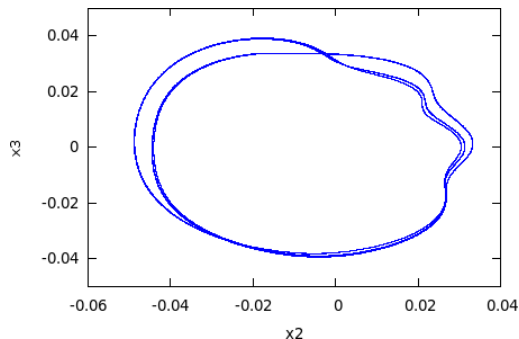
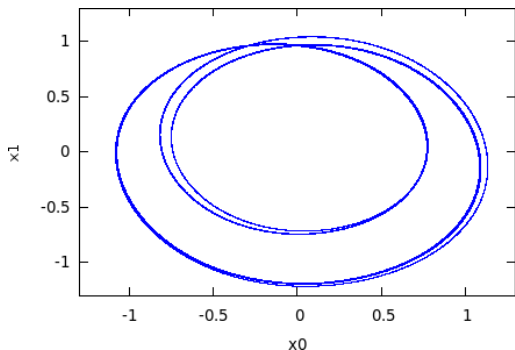
- 1 $P(\mathcal{B}) \subset \mathcal{B}$,
- 2 P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i \geq 0} P^i(\mathcal{B})$
- 3 $\mathcal{A}$ is a nontrivial continuum

Theorem

Put $P(x) = \varphi(T, x)$, where $T = 6$ – period of vector field

There exists compact, connected, explicit set $\mathcal{B}$ ($\sim 8 \cdot 10^6$ boxes) such that

- 1 $P(\mathcal{B}) \subset \mathcal{B}$,
- 2 P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i \geq 0} P^i(\mathcal{B})$
- 3 $\mathcal{A}$ is a nontrivial continuum

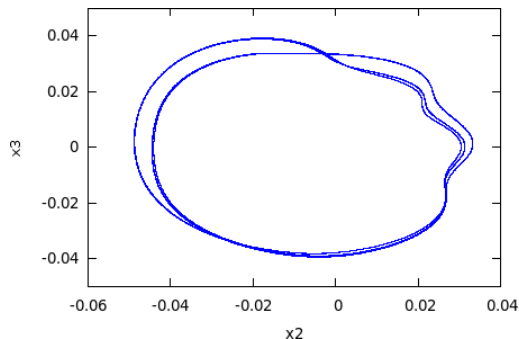
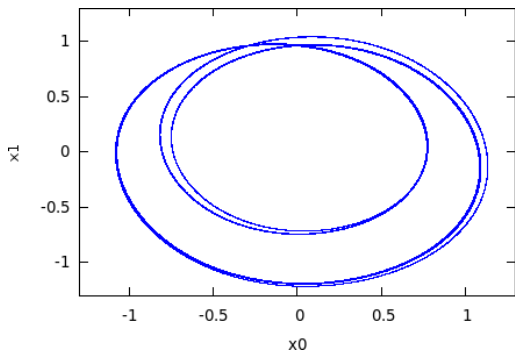


Theorem

Put $P(x) = \varphi(T, x)$, where $T = 6$ – period of vector field

There exists compact, connected, explicit set $\mathcal{B}$ ($\sim 8 \cdot 10^6$ boxes) such that

- 1 $P(\mathcal{B}) \subset \mathcal{B}$,
- 2 P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i \geq 0} P^i(\mathcal{B})$
- 3 $\mathcal{A}$ is a nontrivial continuum

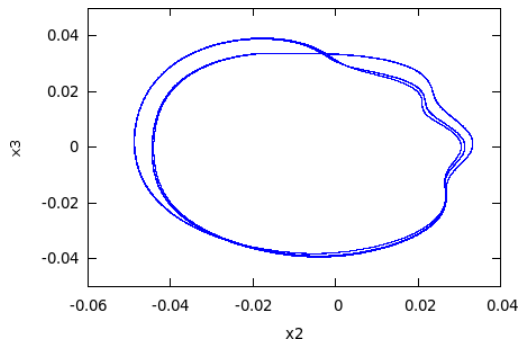
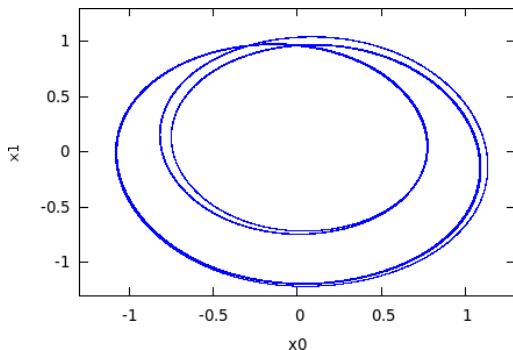


Theorem

Put $P(x) = \varphi(T, x)$, where $T = 6$ – period of vector field

There exists compact, connected, explicit set $\mathcal{B}$ ($\sim 8 \cdot 10^6$ boxes) such that

- 1 $P(\mathcal{B}) \subset \mathcal{B}$,
- 2 P is uniformly hyperbolic on $\mathcal{A} = \bigcap_{i \geq 0} P^i(\mathcal{B})$
- 3 $\mathcal{A}$ is a nontrivial continuum



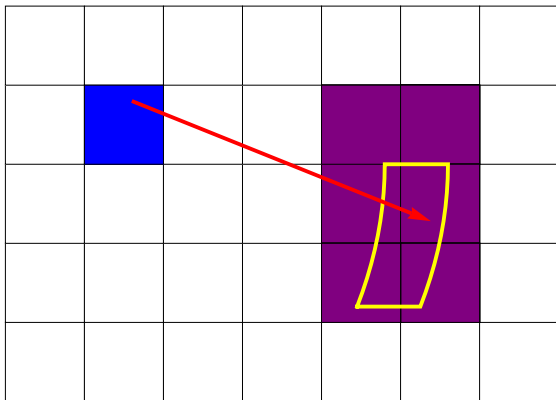
Graph representation of map

$f: D \subset N \rightarrow N$ - a map

N_i - (usually) boxes covering N

Construct **directed graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that

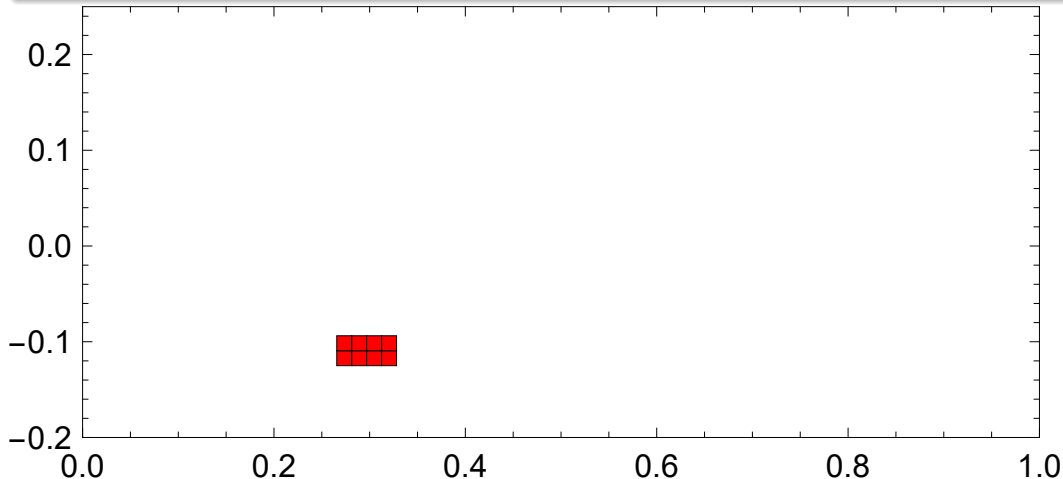
$$f(N_i \cap D) \cap N_j \neq \emptyset \implies (i, j) \in \mathcal{E}$$



Example

Example: Rössler map

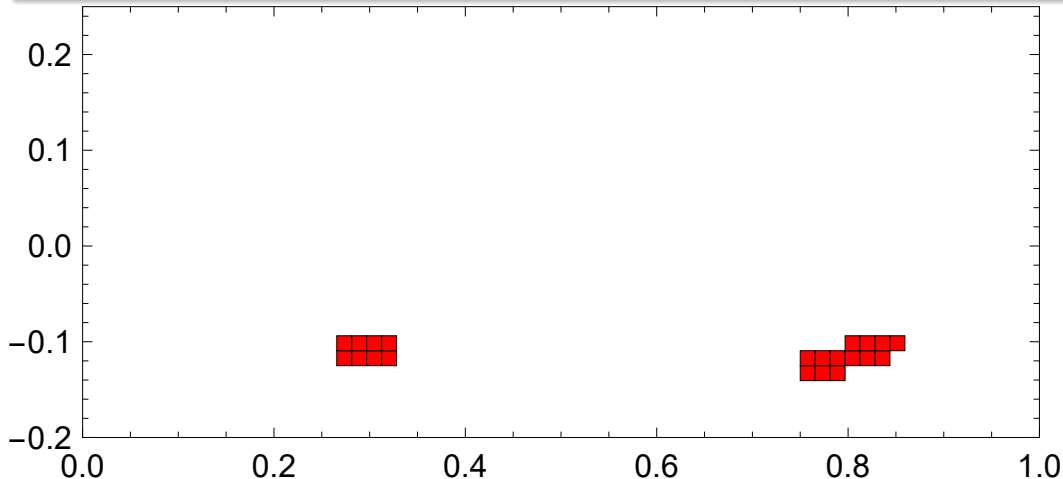
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

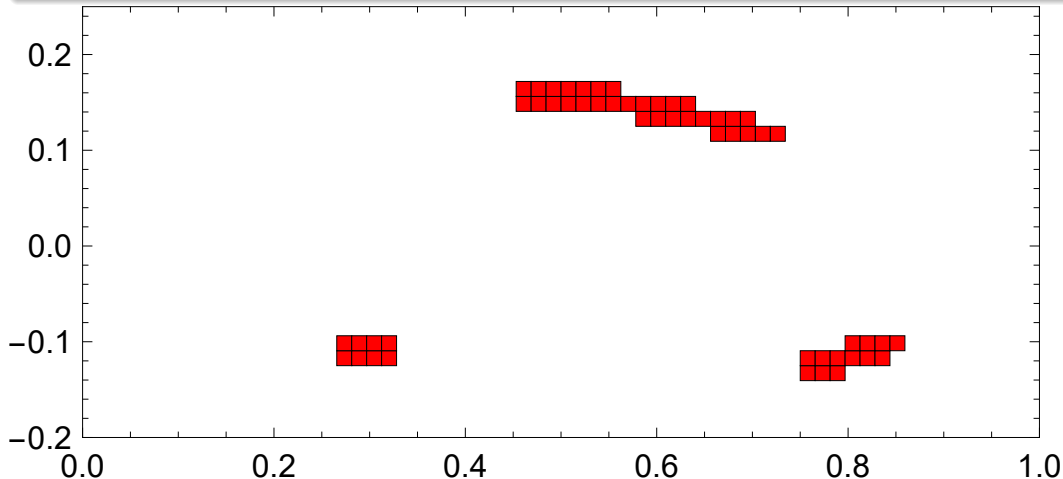
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

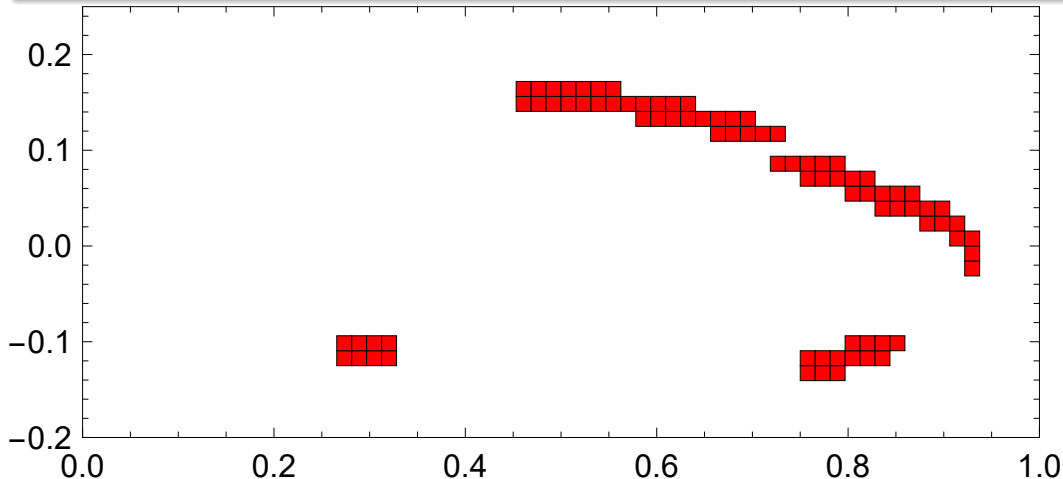
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

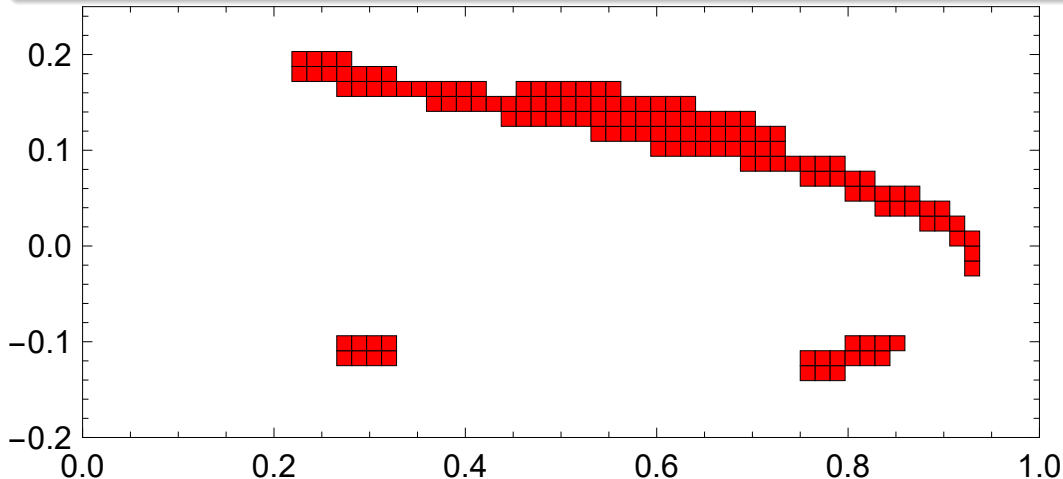
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

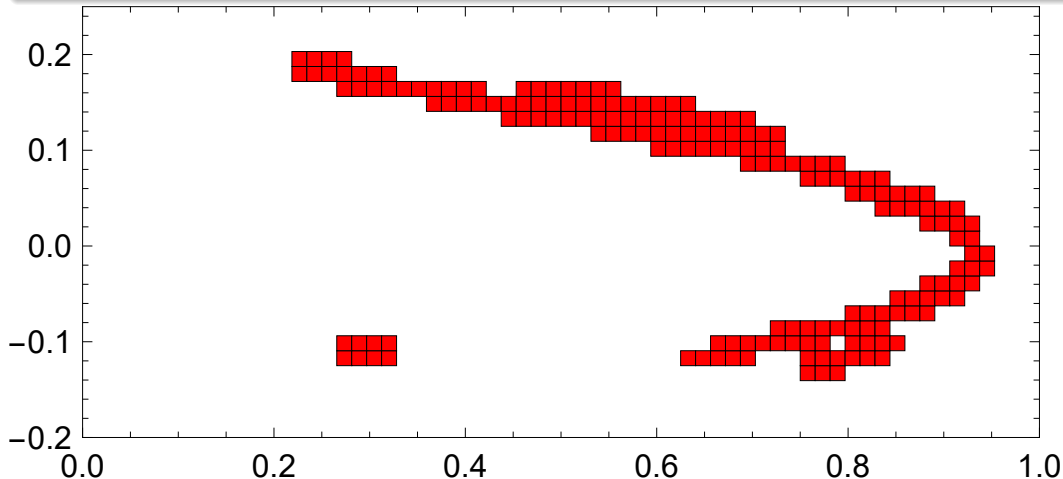
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

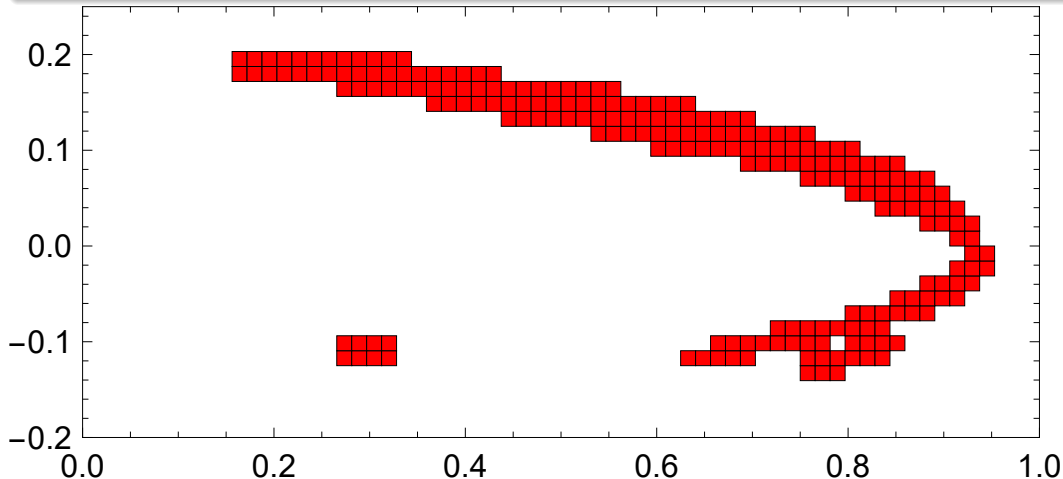
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

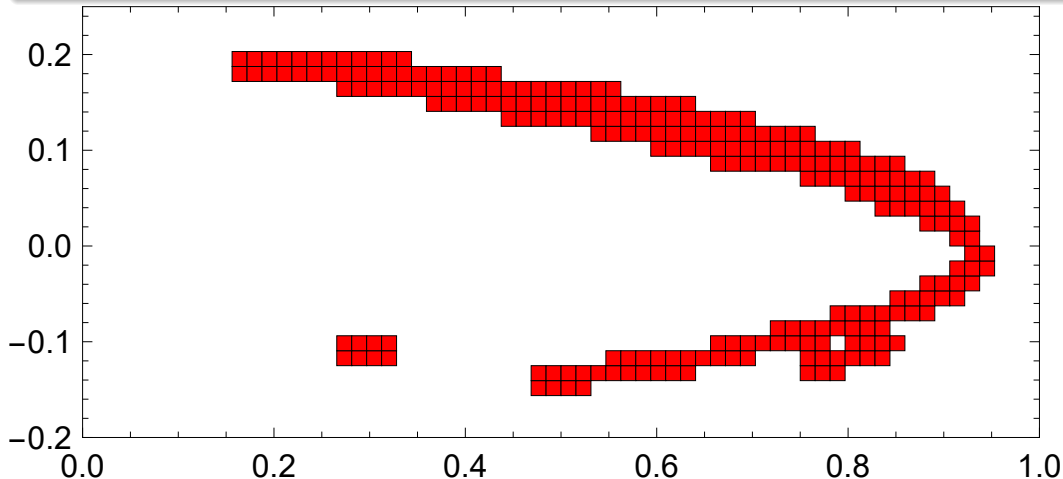
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

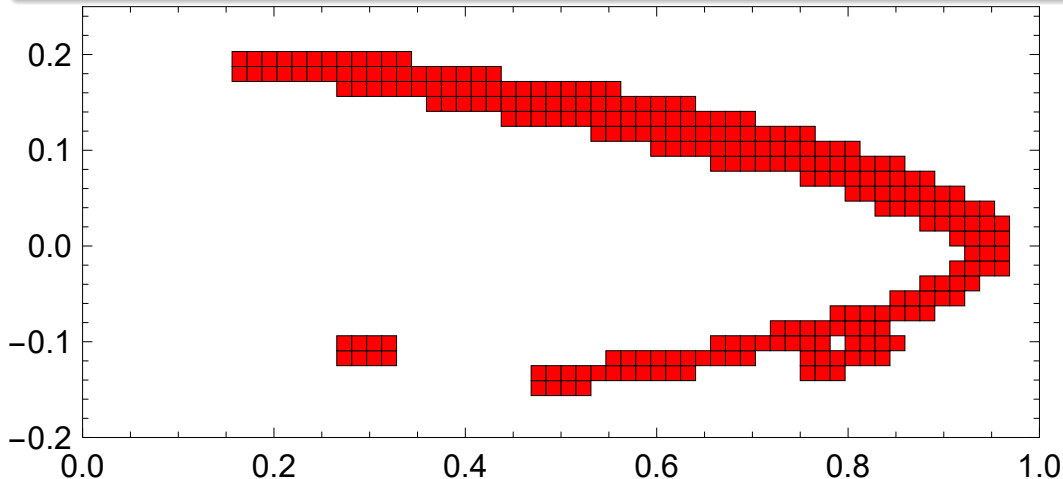
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

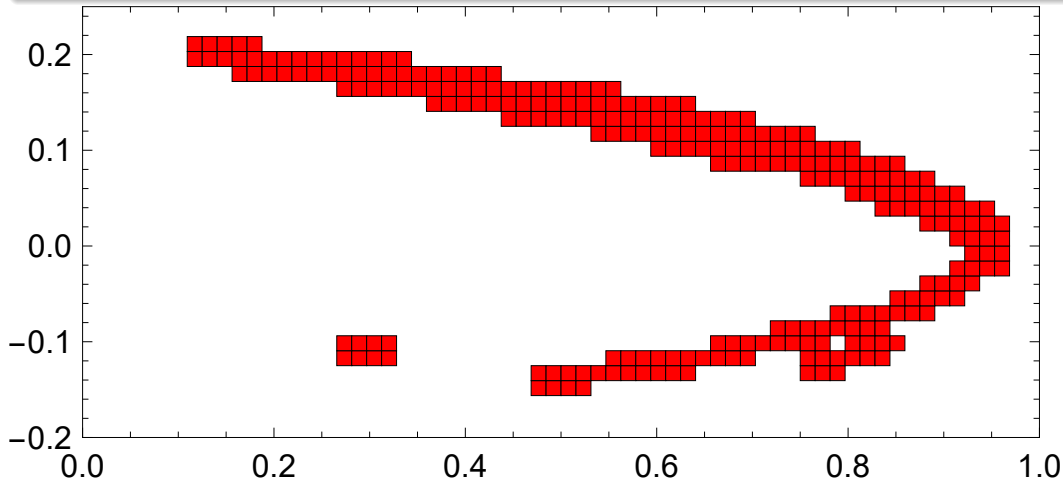
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

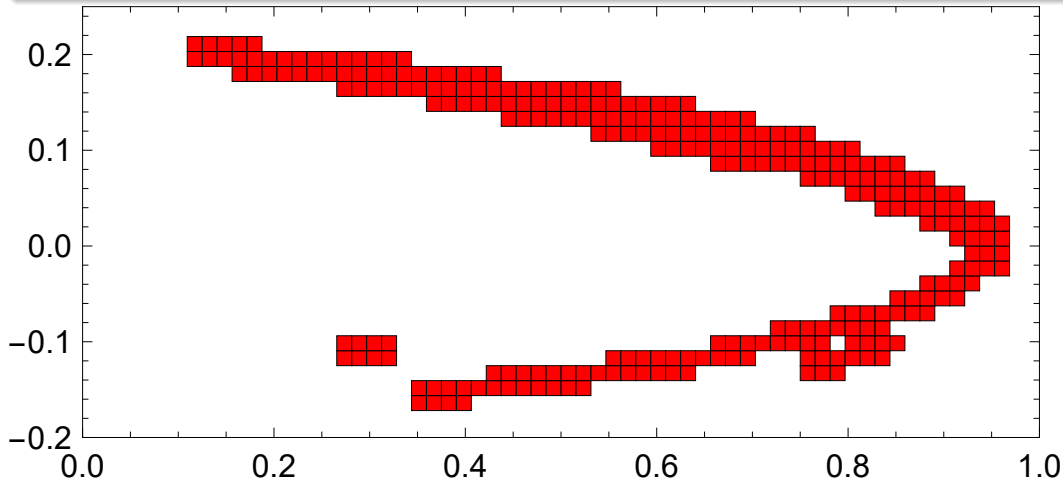
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

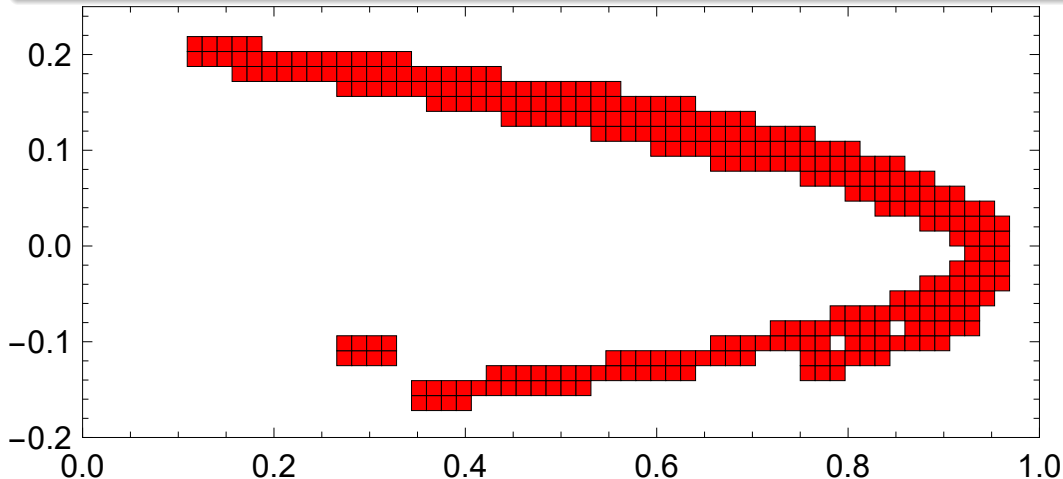
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

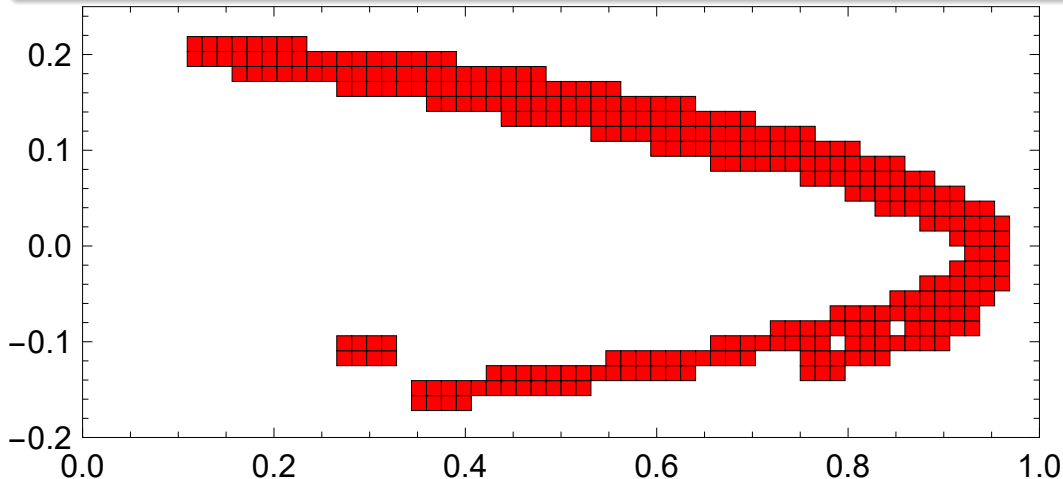
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

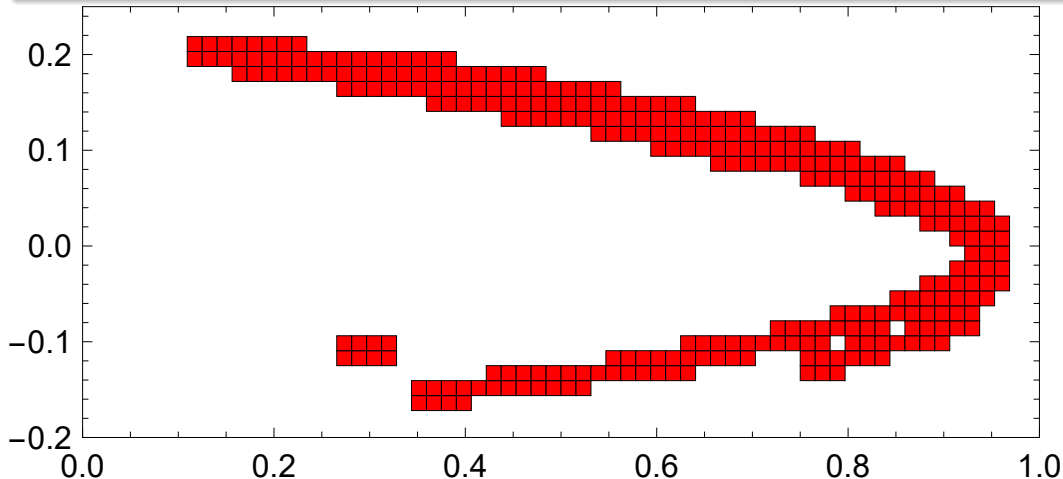
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

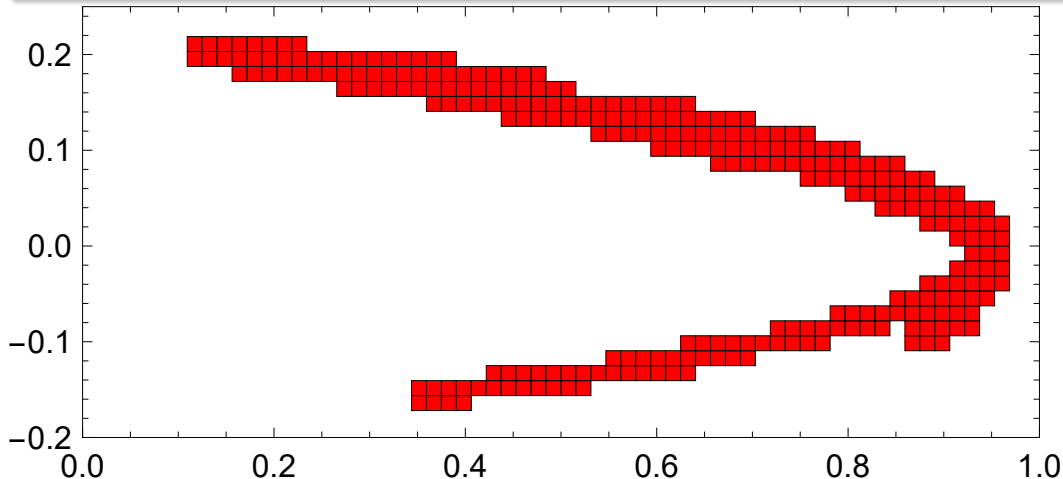
$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$



Example

Example: Rössler map

$$\mathcal{R}(x, y) = (3.8x(1 - x) - 0.1y, \quad 0.2(y - 1.2)(1 - 1.9x))$$

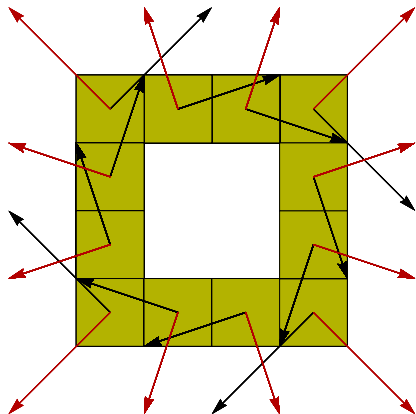


General settings

- $M = \bigcup_{i=1}^N M_i$, where $M_i \subset \mathbb{R}^n$ are compact
- u, s - nonnegative, such that $n = u + s$
- C_i - coordinates in 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.

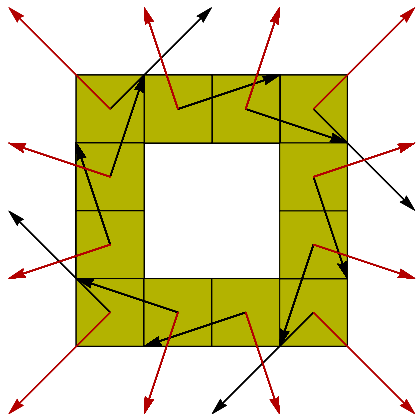


General settings

- $M = \bigcup_{i=1}^N M_i$, where $M_i \subset \mathbb{R}^n$ are compact
- u, s - nonnegative, such that $n = u + s$
- C_i - coordinates in 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.

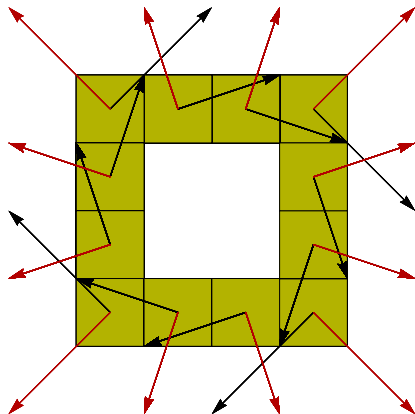


General settings

- $M = \bigcup_{i=1}^N M_i$, where $M_i \subset \mathbb{R}^n$ are compact
- u, s - nonnegative, such that $n = u + s$
- C_i - coordinates in 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.

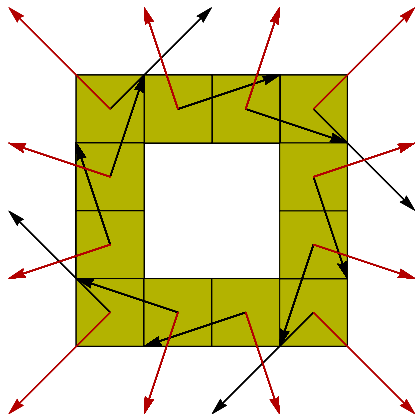


General settings

- $M = \bigcup_{i=1}^N M_i$, where $M_i \subset \mathbb{R}^n$ are compact
- u, s - nonnegative, such that $n = u + s$
- C_i - coordinates in 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.

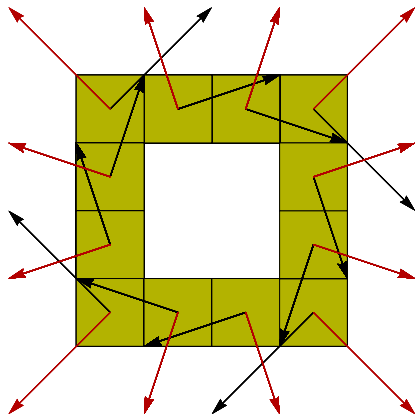


General settings

- $M = \bigcup_{i=1}^N M_i$, where $M_i \subset \mathbb{R}^n$ are compact
- u, s - nonnegative, such that $n = u + s$
- C_i - coordinates in M_i
- quadratic form on $\mathbb{R}^n$

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

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

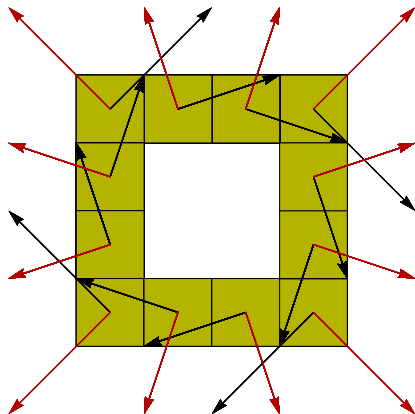


General settings

- $M = \bigcup_{i=1}^N M_i$, where $M_i \subset \mathbb{R}^n$ are compact
- u, s - nonnegative, such that $n = u + s$
- C_i - coordinates in M_i
- quadratic form on $\mathbb{R}^n$

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

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



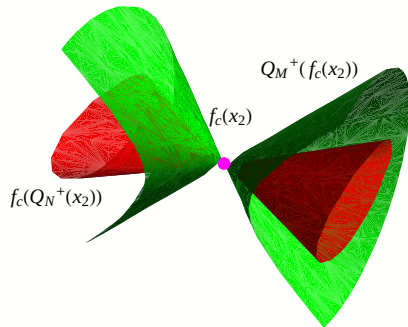
Definition

f is **strongly hyperbolic** on

$\mathcal{M} = (Q, \{(M_i, C_i)\}_{i=1}^N)$ if for $z \in M_i$ and $j = 1, \dots, N$ 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.



Theorem

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



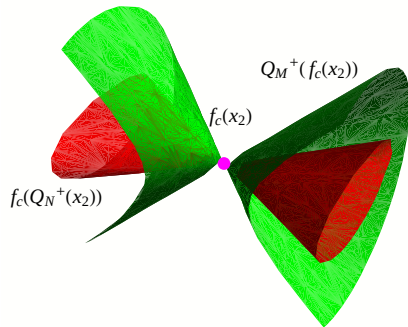
f is uniformly hyperbolic on $\mathcal{H} = \text{Inv}(f, M)$.

Definition

f is **strongly hyperbolic** on $\mathcal{M} = (Q, \{(M_i, C_i)\}_{i=1}^N)$ if for $z \in M_i$ and $j = 1, \dots, N$ 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.



Theorem

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



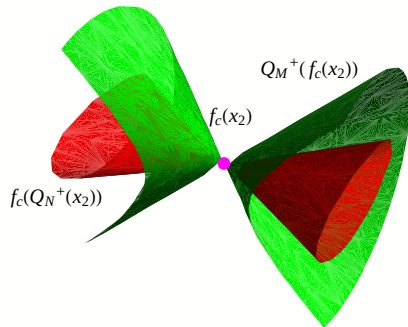
f is uniformly hyperbolic on $\mathcal{H} = \text{Inv}(f, M)$.

Definition

f is **strongly hyperbolic** on $\mathcal{M} = (Q, \{(M_i, C_i)\}_{i=1}^N)$ if for $z \in M_i$ and $j = 1, \dots, N$ 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.



Theorem

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



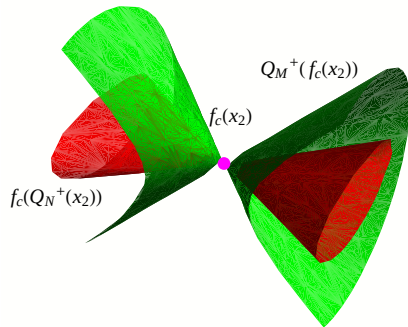
f is uniformly hyperbolic on $\mathcal{H} = \text{Inv}(f, M)$.

Definition

f is **strongly hyperbolic** on $\mathcal{M} = (Q, \{(M_i, C_i)\}_{i=1}^N)$ if for $z \in M_i$ and $j = 1, \dots, N$ 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.



Theorem

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



f is uniformly hyperbolic on $\mathcal{H} = \text{Inv}(f, M)$.

Hyperbolic attractor in the Kuznetsov system

Algorithm	Wall time (h:mm)	Comments
enclosure	2:16, 224 CPUs	7 970 392 boxes
cycles in graph, max period 6	0:58, 32 CPUs	2190 cycles
periodic points, max period 6	0:06, 32 CPUs	105 POs
coord systems in boxes	6:59, 32 CPUs	
strong hyperbolicity	4:24, 224 CPUs	$\mathcal{C}^1$ -algorithm

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

Thank you for your attention!

<https://github.com/CAPDgroup>