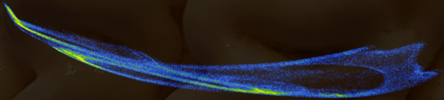
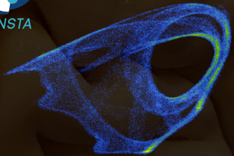
# Nonlinear Oscillator: An Attempt to Predict Butterfly Effect by Machine learning

Vladislav A. YASTREBOV[1], Yirun ZOU[1,2]

[1]*MINES ParisTech, PSL University, Centre des Matériaux, CNRS UMR 7633, Evry*

[2]*ENSTA, Palaiseau*





my strange attractor

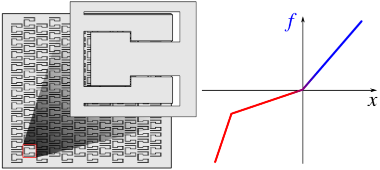
*Big SIMS*

*Evry, France*

*September 10, 2019*

# Outline

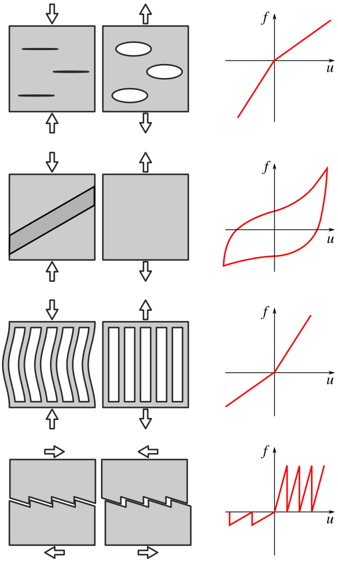
# Introduction



# Asymmetry in materials

**Sources of asymmetry**[A,B]:

- Micro-fractures
  *concrete/rocks*
- Local buckling/wrinkling
  *fiber networks, ropes, membranes*
- Phase transformations/twinning
  *Mg alloys, Ti alloys*
- Pressure dependent plasticity
  *concrete/rocks/soils*
- Sliding asymmetry[C]
  *patterned surfaces*
- ★ Contacts
  *granular matter, granular crystals*





[A] Ambartsumyan (1965), Izv Academ Nauk USSR Mech

[B] Gibson, Ashby (1987), Cellular Solids, Cambridge UP

[C] Rafsanjani, Bertoldi, et al (2018), Science Robotics

# Asymmetry in materials

**Sources of asymmetry**[A,B]:

- Micro-fractures
  *concrete/rocks*
- Local buckling/wrinkling
  *fiber networks, ropes, membranes*
- Phase transformations/twinning
  *Mg alloys, Ti alloys*
- Pressure dependent plasticity
  *concrete/rocks/soils*
- Sliding asymmetry[C]
  *patterned surfaces*
- ★ Contacts
  *granular matter, granular crystals*

[A] Ambartsumyan (1965), Izv Academ Nauk USSR Mech
[B] Gibson, Ashby (1987), Cellular Solids, Cambridge UP
[C] Rafsanjani, Bertoldi, et al (2018), Science Robotics

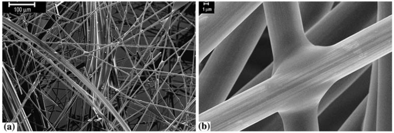



Fig. Carbon fiber network
[1] Mezeix, Bouvet, Huez, Poquillon (2009). J Mater Sci 44

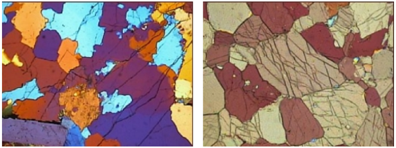


Fig. Microcracks in rocks (dolomite, granite)
[2] Obara (2007). Comput & Geosci 33
[3] Obara, Kozusnikova (2007). Computat Geosci 11

Fig. Torsional instability in multi-strand wires (ropes)
[4] www.industrialrope.com

# Asymmetry in materials

**Sources of asymmetry**[A,B]:

- Micro-fractures
  *concrete/rocks*
- Local buckling/wrinkling
  *fiber networks, ropes, membranes*
- Phase transformations/twinning
  *Mg alloys, Ti alloys*
- Pressure dependent plasticity
  *concrete/rocks/soils*
- Sliding asymmetry[C]
  *patterned surfaces*
- ★ Contacts
  *granular matter, granular crystals*

[A] Ambartsumyan (1965), Izv Academ Nauk USSR Mech

[B] Gibson, Ashby (1987), Cellular Solids, Cambridge UP

[C] Rafsanjani, Bertoldi, et al (2018), Science Robotics

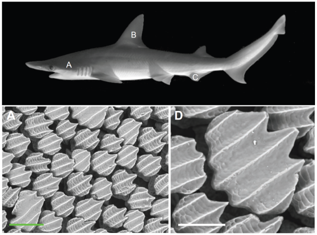



Fig. Shark skin
[5] Wen, Weaver, Lauder (2014). J Exp Biol 217

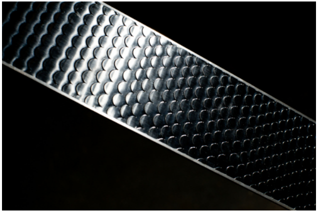
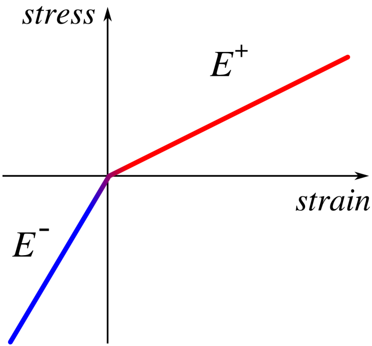


Fig. Fish-skin pattern on skitour skis
[6] Photo courtesy "Voile"

# Bibliographical sketch

**Hetero-modulus, multi-modulus, bi-modulus**

- Sergey A. Ambartsumyan (1965-1969, 1982)
- Robert M. Jones (1971,1975, 1977)
- Alain Curnier, Qi-Chang He, Philippe Zysset (1995)



# Bibliographical sketch

**Hetero-modulus, multi-modulus, bi-modulus**

- Sergey A. Ambartsumyan (1965-1969, 1982)
- Robert M. Jones (1971,1975, 1977)
- Alain Curnier, Qi-Chang He, Philippe Zysset (1995)

Elastic properties depend on principal stresses

$$\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\end{bmatrix} = \begin{bmatrix}S_{11}(\sigma_1) & S_{12}(\sigma_2) & S_{13}(\sigma_3)\\ S_{21}(\sigma_1) & S_{22}(\sigma_2) & S_{23}(\sigma_3)\\ S_{31}(\sigma_1) & S_{32}(\sigma_2) & S_{33}(\sigma_3)\end{bmatrix}\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\end{bmatrix}$$

$$S_{11}(+) = S_{22}(+) = S_{33}(+) = \frac{1}{E^+}, \qquad S_{11}(-) = S_{22}(-) = S_{33}(-) = \frac{1}{E^-}$$

$$S_{12}(+) = S_{13}(+) = S_{23}(+) = -\frac{\nu^+}{E^+}, \qquad S_{12}(-) = S_{13}(-) = S_{23}(-) = -\frac{\nu^-}{E^-}$$

For symmetry

$$\nu^- E^+ = \nu^+ E^-$$

- Adapted variational framework[1].
- Anisotropic damage[2].

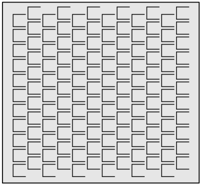
[1] Du, Guo (2014). J Mech Phys Solids 73
[2] Desmorat, Gatuingt, Ragueneau (2007). Eng Fract Mech 74

# Contact-based architectured material

- Contact as a functional element

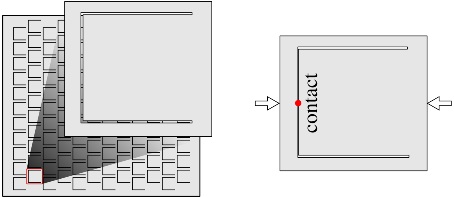
# Contact-based architectured material

- Contact as a functional element
- Stiff in compression / soft in tension



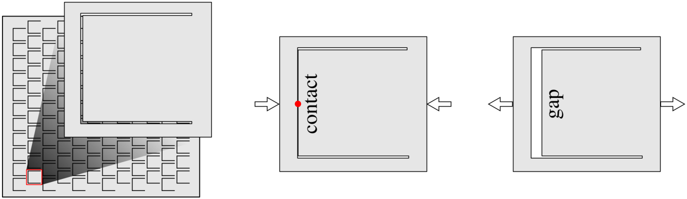
# Contact-based architectured material

- Contact as a functional element
- Stiff in compression / soft in tension



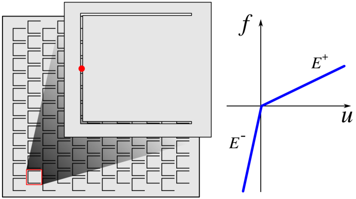
# Contact-based architectured material

- Contact as a functional element
- Stiff in compression / soft in tension



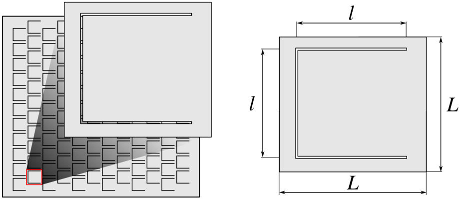
# Contact-based architectured material

- Contact as a functional element
- Stiff in compression / soft in tension



# Contact-based architectured material

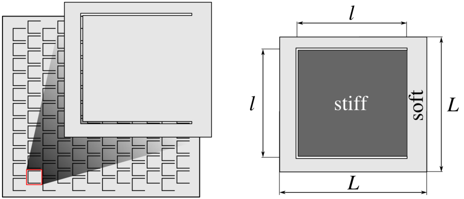
- Contact as a functional element
- Stiff in compression / soft in tension




$$E^{+}/E^{-} \approx 1 - l/L$$

# Contact-based architectured material

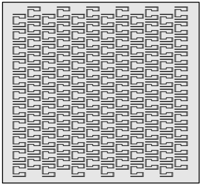
- Contact as a functional element
- Stiff in compression / soft in tension
- Use different materials




$$E^+/E^- \approx (1 - l/L)\ E^{\text{soft}}/E^{\text{stiff}}$$

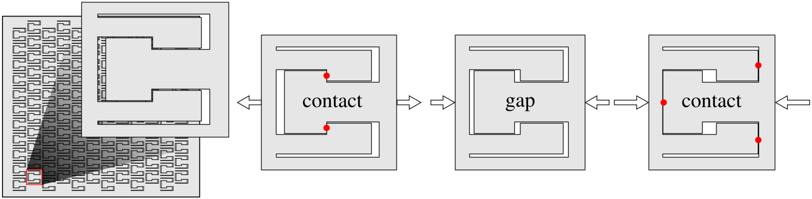
# Contact-based architectured material

- Contact as a functional element
- Stiff in compression / soft in tension
- Use different materials
- Soft in compression/stiff in tension



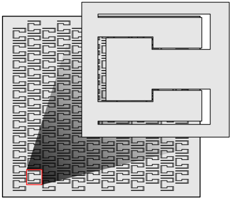
# Contact-based architectured material

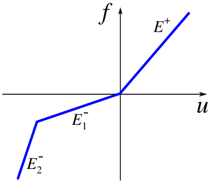
- Contact as a functional element
- Stiff in compression / soft in tension
- Use different materials
- Soft in compression/stiff in tension



# Contact-based architectured material

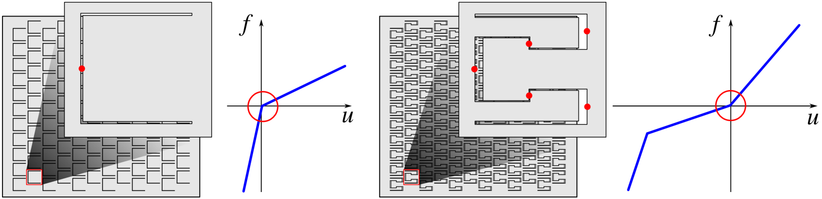
- Contact as a functional element
- Stiff in compression / soft in tension
- Use different materials
- Soft in compression/stiff in tension





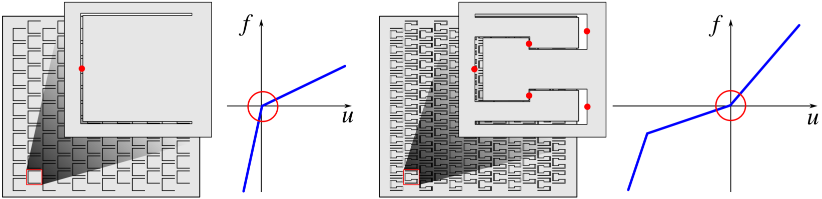
# Contact-based architectured material

- Contact as a functional element
- Stiff in compression / soft in tension
- Use different materials
- Soft in compression/stiff in tension



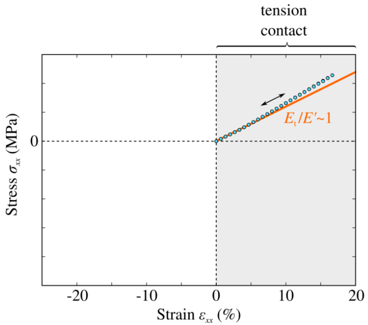
# Contact-based architectured material

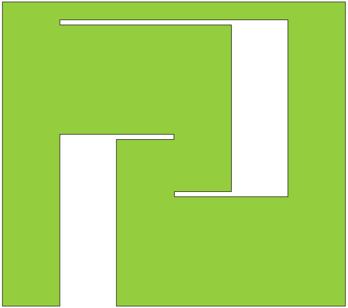
- Contact as a functional element
- Stiff in compression / soft in tension
- Use different materials
- Soft in compression/stiff in tension
- Extendable to 3D



# Stiff-in-tension / soft-in-compression: example

- Tension/compression test (FE simulation)



# Governing equation for asymmetric materials

- Quastistatic 1D behavior:

$$\sigma = E(\nabla u + \alpha|\nabla u|) = \begin{cases} (1+\alpha)E\nabla u, & \text{if } \nabla u > 0, \text{ tension} \\ (1-\alpha)E\nabla u, & \text{if } \nabla u \le 0, \text{ compres.} \end{cases}, \quad -1 < \alpha < 1$$

- Elastic contrast $\gamma$:

$$\frac{E^+}{E^-} = \frac{1+\alpha}{1-\alpha} = \gamma$$

- Elastodynamic equation in 1D (the simplest approximation):

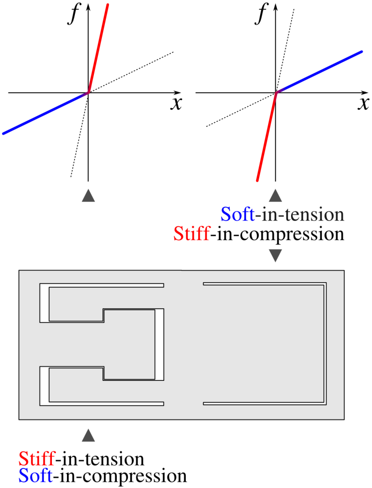
$$\rho\ddot{u} = E\nabla\left(\nabla u + \alpha|\nabla u|\right) + \underbrace{\mu\Delta\dot{u}}_{\text{damping}}$$

- Wave celerity $c$:

$$c = \begin{cases} \sqrt{(1+\alpha)E/\rho}, & \text{if } \nabla u > 0, \text{ tension} \\ \sqrt{(1-\alpha)E/\rho}, & \text{if } \nabla u \le 0, \text{ compres.} \end{cases}, \quad \frac{c^+}{c^-} = \sqrt{\frac{1+\alpha}{1-\alpha}} = \sqrt{\gamma}$$

# Set-up and simplified model

- Adjust elastic properties
- Combine in a quasi-statically symmetric element



# Set-up and simplified model

- Adjust elastic properties
- Combine in a quasi-statically symmetric element
- Simplify

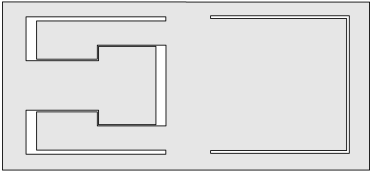
$$k^{soft} = \frac{E^{soft}A}{L}, \quad k^{stiff} = \frac{E^{stiff}A}{L}, \quad \gamma = \frac{k^{stiff}}{k^{soft}}$$

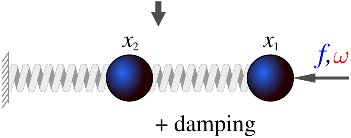
$$k^{eff}_{spring} = \frac{2\gamma}{1+\gamma}k^{soft}$$

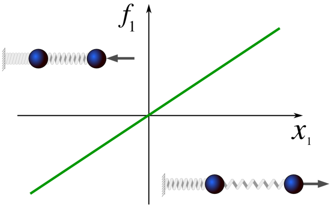
$$\boxed{\ddot{X} + 2\eta C\dot{X} + k^{soft}K(X)X = F}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} sin(\omega t + \phi) \\ 0 \end{bmatrix},$$

$$K = \begin{bmatrix} \beta(x_1 - x_2) & -\beta(x_1 - x_2) \\ -\beta(x_1 - x_2) & \beta(x_1 - x_2) + \beta(-x_2) \end{bmatrix}$$





# Set-up and simplified model

- Adjust elastic properties
- Combine in a quasi-statically symmetric element
- Simplify

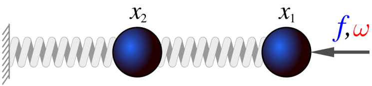
$$k^{soft} = \frac{E^{soft}A}{L}, \quad k^{stiff} = \frac{E^{stiff}A}{L}, \quad \gamma = \frac{k^{stiff}}{k^{soft}}$$

$$k^{eff}_{spring} = \frac{2\gamma}{1+\gamma} k^{soft}$$

$$\boxed{\ddot{X} + 2\eta C\dot{X} + k^{soft} K(X) X = F}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} sin(\omega t + \phi) \\ 0 \end{bmatrix},$$

$$K = \begin{bmatrix} \beta(x_1 - x_2) & -\beta(x_1 - x_2) \\ -\beta(x_1 - x_2) & \beta(x_1 - x_2) + \beta(-x_2) \end{bmatrix}$$




+ damping

$$\beta(x) = \begin{cases} \gamma, & \text{if } x < 0; \\ 1, & \text{if } x \geq 0. \end{cases}$$

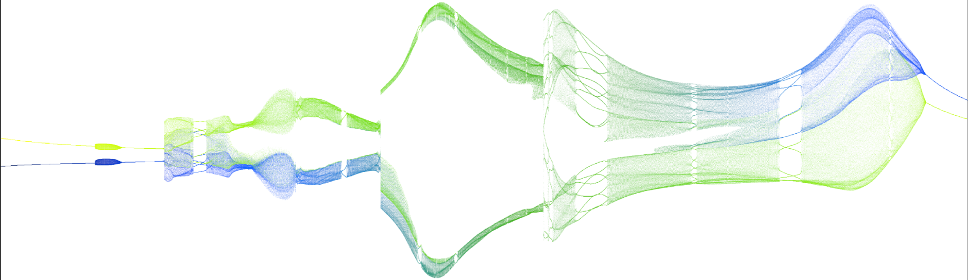
- Tens-tens / comp-comp

$$K_{tt} = \begin{bmatrix} 1 & -1 \\ -1 & 1+\gamma \end{bmatrix}, \quad K_{cc} = \begin{bmatrix} \gamma & -\gamma \\ -\gamma & 1+\gamma \end{bmatrix}$$

- Tens-comp / comp-tens

$$K_{tc} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}, \quad K_{ct} = \begin{bmatrix} \gamma & -\gamma \\ -\gamma & 2\gamma \end{bmatrix},$$

# Forced oscillations



# Analysis principle

**System of equations**

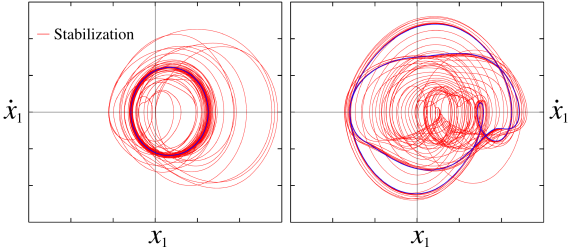
$$\ddot{X} + 2\eta C\dot{X} + K(X,\gamma)X = F(\omega)$$

**Parameters:**

- damping $\eta \in [0.01; 0.2]$ in 10 steps
- stiffness contrast $\gamma \in [1; 5]$ in 800 steps
- forcing frequency $\omega \in [0.1, 7.0]$ in 10 000 steps

**Track:**

- Stable cycle




*Poincaré point: $\{x_1, \dot{x}_1\}$ for $\omega t = 2\pi n$

# Analysis principle

**System of equations**

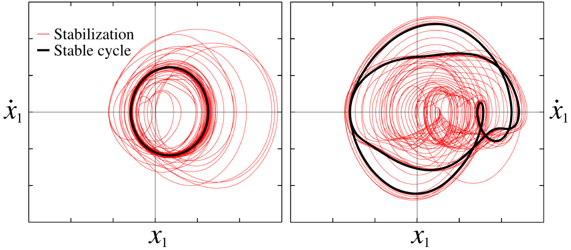
$$\ddot{X} + 2\eta C\dot{X} + K(X,\gamma)X = F(\omega)$$

**Parameters:**

- damping $\eta \in [0.01; 0.2]$ in 10 steps
- stiffness contrast $\gamma \in [1; 5]$ in 800 steps
- forcing frequency $\omega \in [0.1, 7.0]$ in 10 000 steps

**Track:**

- Stable cycle
- Min/max amplitude




*Poincaré point: $\{x_1, \dot{x}_1\}$ for $\omega t = 2\pi n$

# Analysis principle

**System of equations**

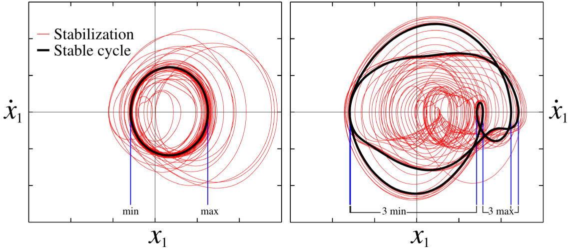
$$\ddot{X} + 2\eta C\dot{X} + K(X,\gamma)X = F(\omega)$$

**Parameters:**

- damping $\eta \in [0.01; 0.2]$ in 10 steps
- stiffness contrast $\gamma \in [1; 5]$ in 800 steps
- forcing frequency $\omega \in [0.1, 7.0]$ in 10 000 steps

**Track:**

- Stable cycle
- Min/max amplitude
- Number of amplitude extrema




*Poincaré point: $\{x_1, \dot{x}_1\}$ for $\omega t = 2\pi n$

# Analysis principle

## System of equations

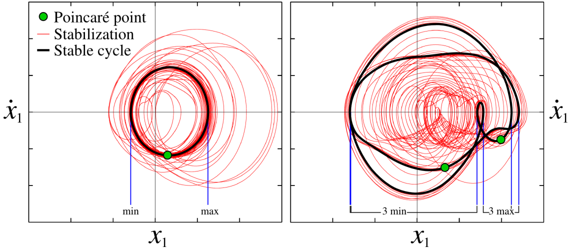
$$\ddot{X} + 2\eta C\dot{X} + K(X,\gamma)X = F(\omega)$$

**Parameters:**

- damping $\eta \in [0.01; 0.2]$ in 10 steps
- stiffness contrast $\gamma \in [1; 5]$ in 800 steps
- forcing frequency $\omega \in [0.1, 7.0]$ in 10 000 steps

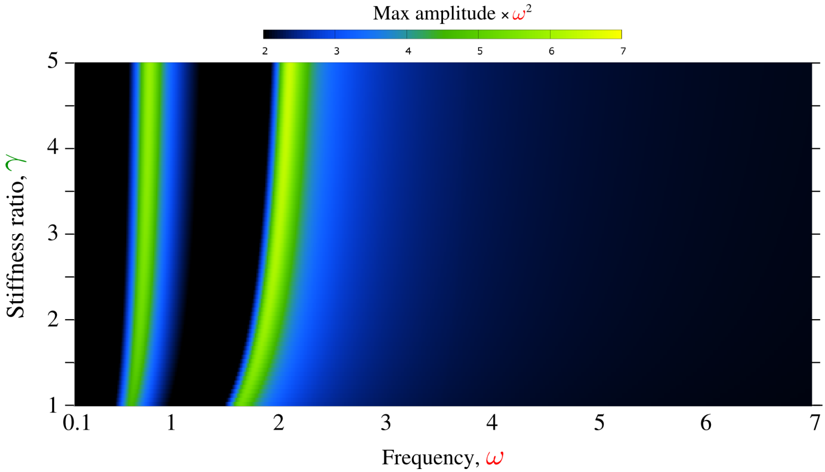
**Track:**

- Stable cycle
- Min/max amplitude
- Number of amplitude extrema
- Number of Poincaré points*



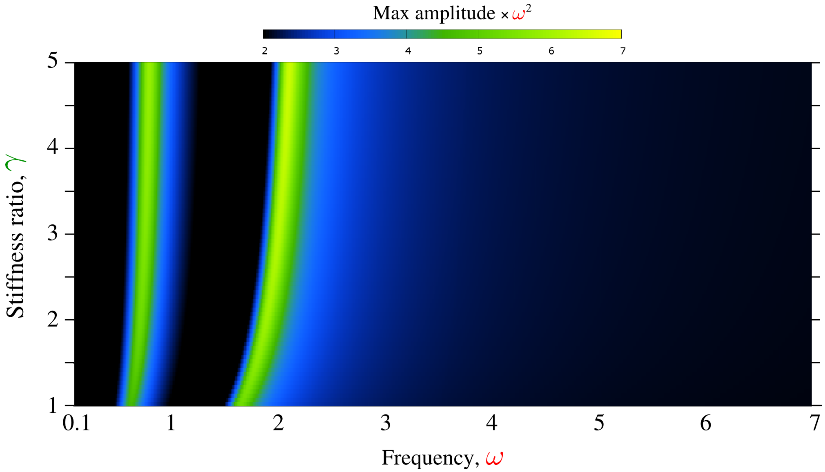

*Poincaré point: $\{x_1, \dot{x}_1\}$ for $\omega t = 2\pi n$

# Homogenized model

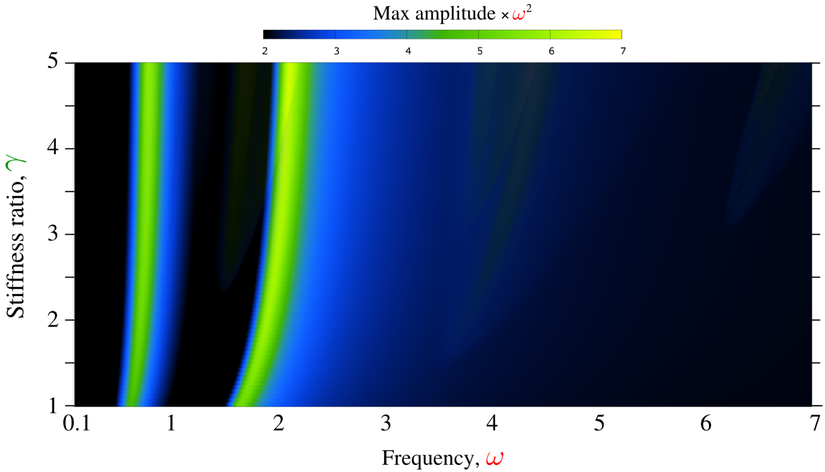



- One maximum, one minimum, one Poincaré point.

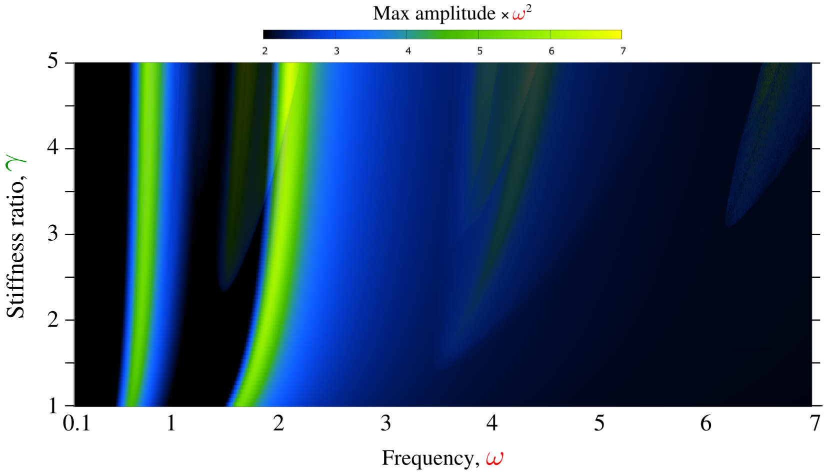
# Bimodulus system $\alpha = 0.1$



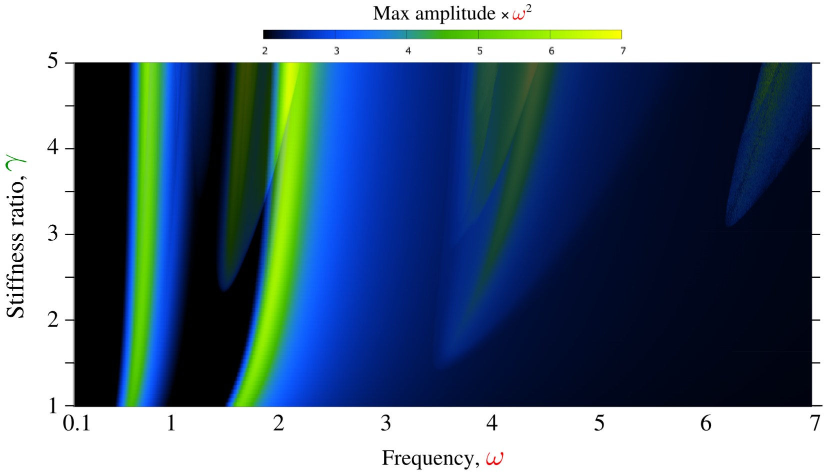
# Bimodulus system $\alpha = 0.1$



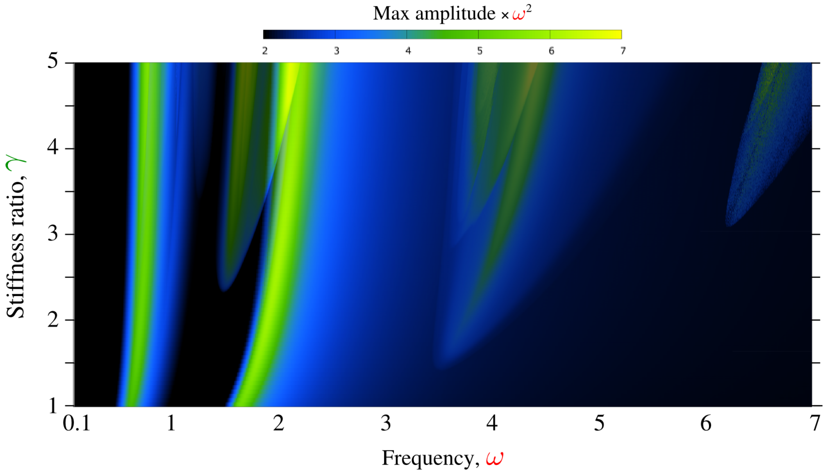
# Bimodulus system $\alpha = 0.1$



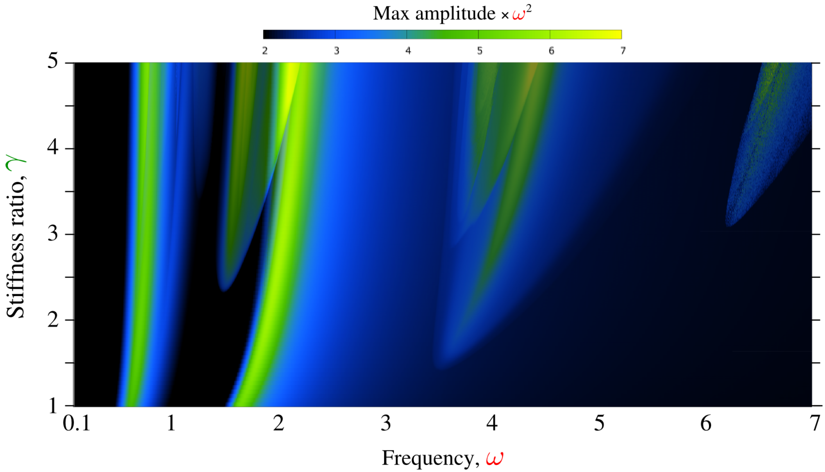
# Bimodulus system $\alpha = 0.1$



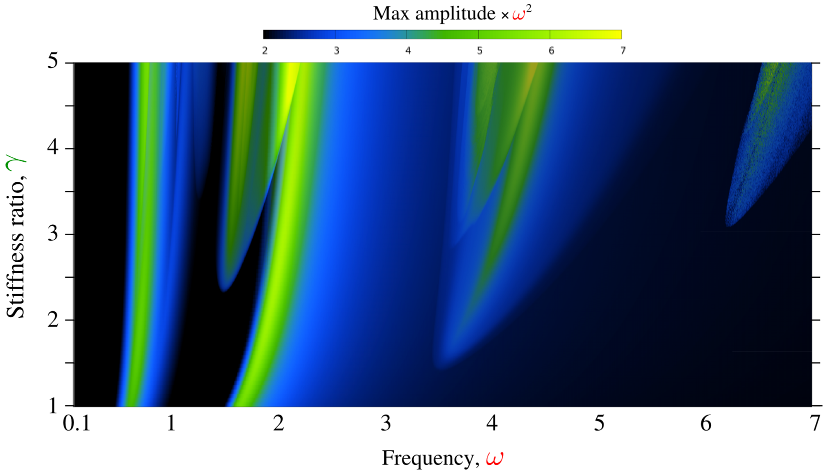
# Bimodulus system $\alpha = 0.1$



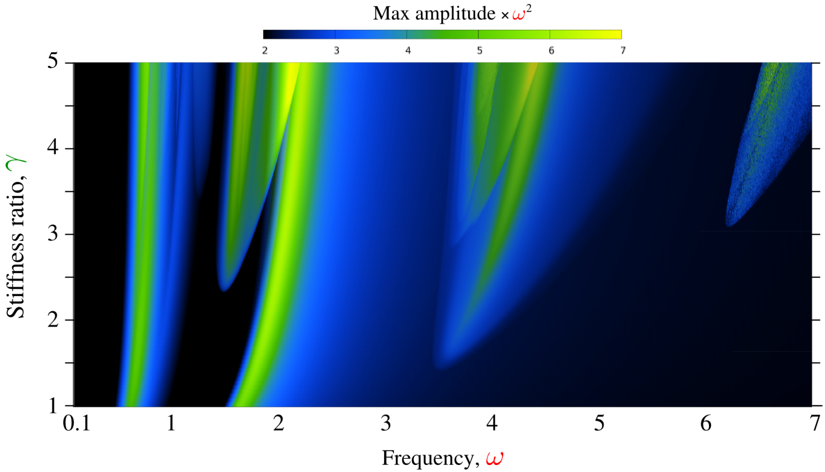
# Bimodulus system $\alpha = 0.1$



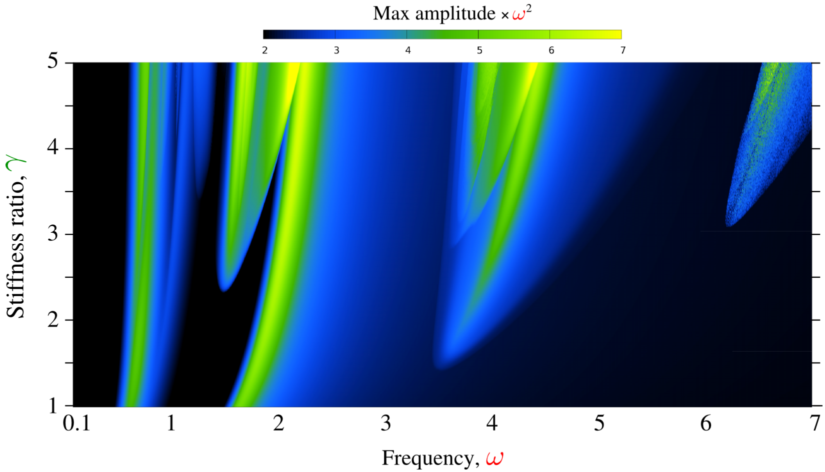
# Bimodulus system $\alpha = 0.1$



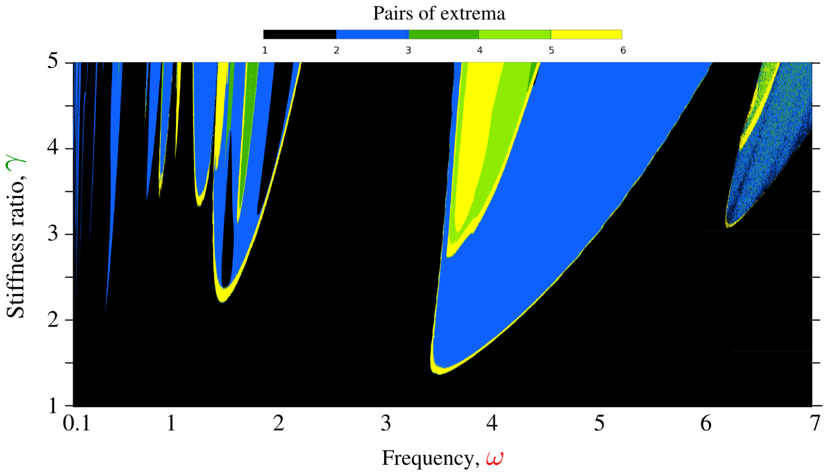
# Bimodulus system $\alpha = 0.1$



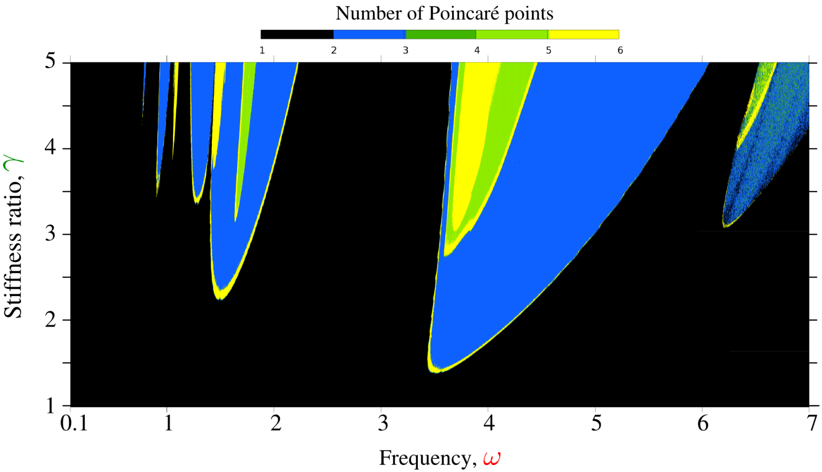
# Bimodulus system $\alpha = 0.1$



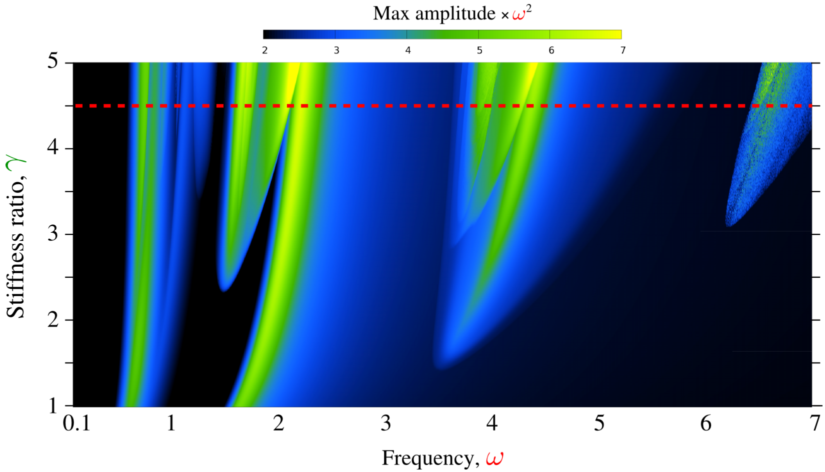
# Bimodulus system $\alpha = 0.1$



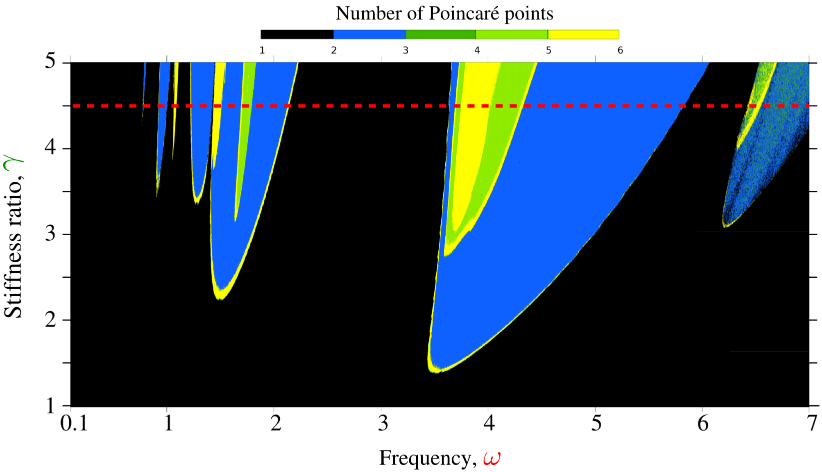
# Bimodulus system $\alpha = 0.1$



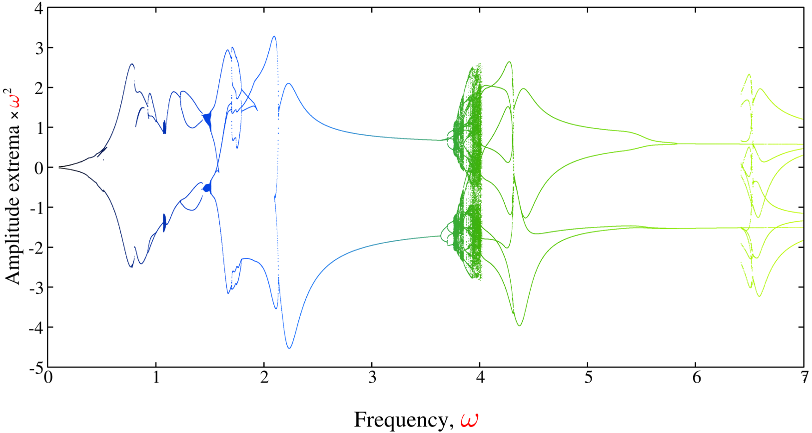
# Bimodulus system $\alpha = 0.1$: dive in the plot



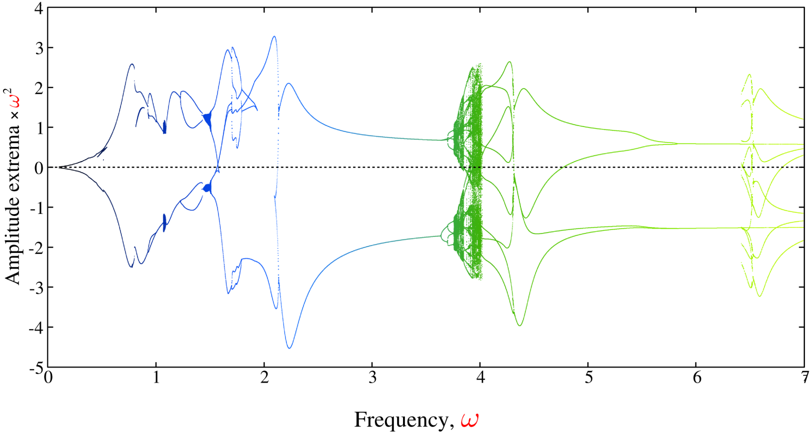
# Bimodulus system $\alpha = 0.1$: dive in the plot



# Bimodulus system $\alpha = 0.1$: dive in the plot

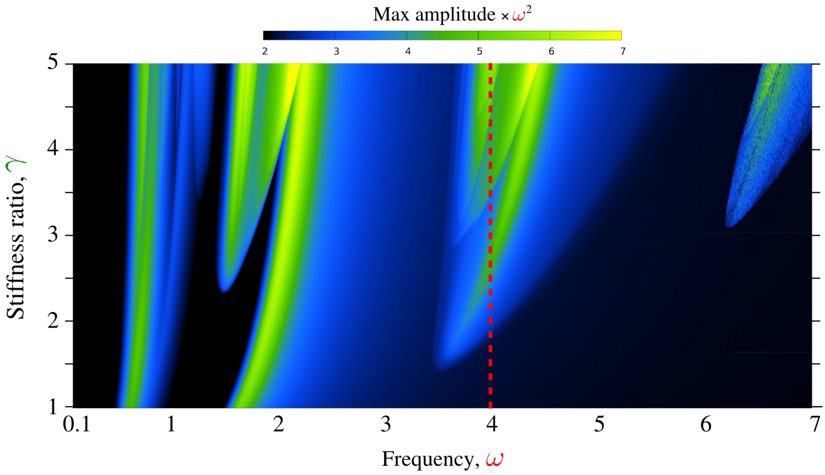


# Bimodulus system $\alpha = 0.1$: dive in the plot

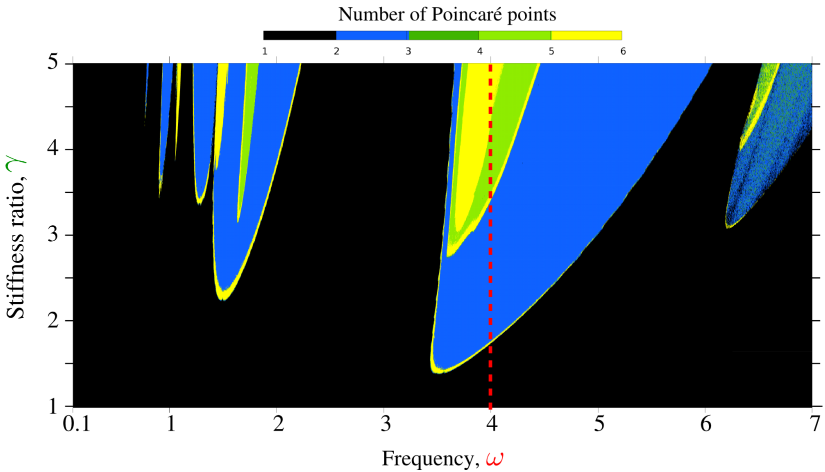



- Asymmetrical oscillations

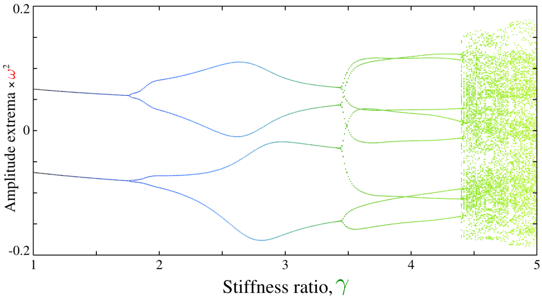
# Bimodulus system $\alpha = 0.1$: dive in the plot II



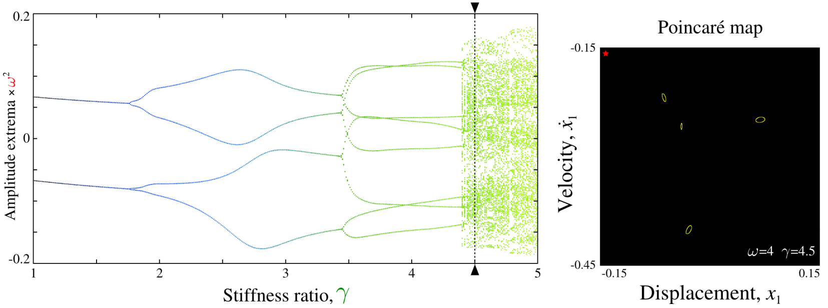
# Bimodulus system $\alpha = 0.1$: dive in the plot II



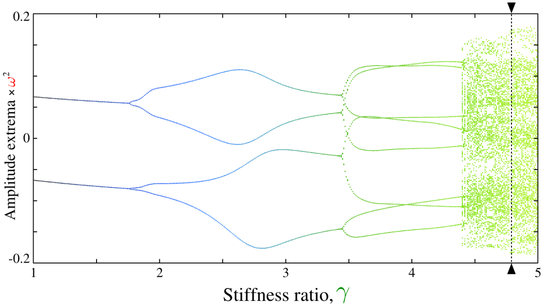
# Bimodulus system $\alpha = 0.1$: dive in the plot II

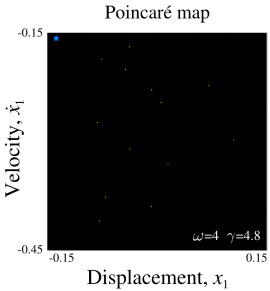


# Bimodulus system $\alpha = 0.1$: dive in the plot II

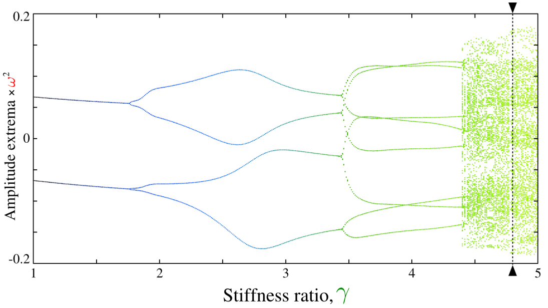


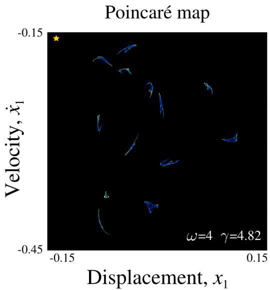
# Bimodulus system $\alpha = 0.1$: dive in the plot II



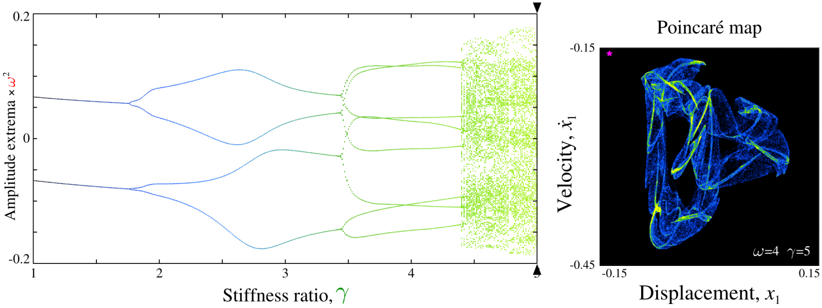



# Bimodulus system $\alpha = 0.1$: dive in the plot II



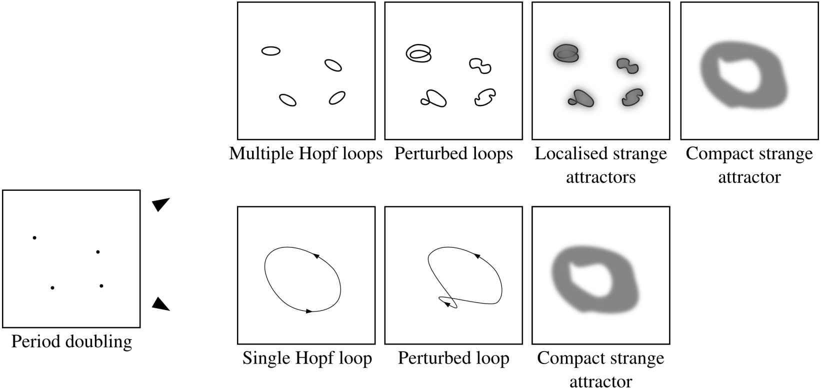
# Bimodulus system $\alpha = 0.1$: dive in the plot II




Vizualise in 4D

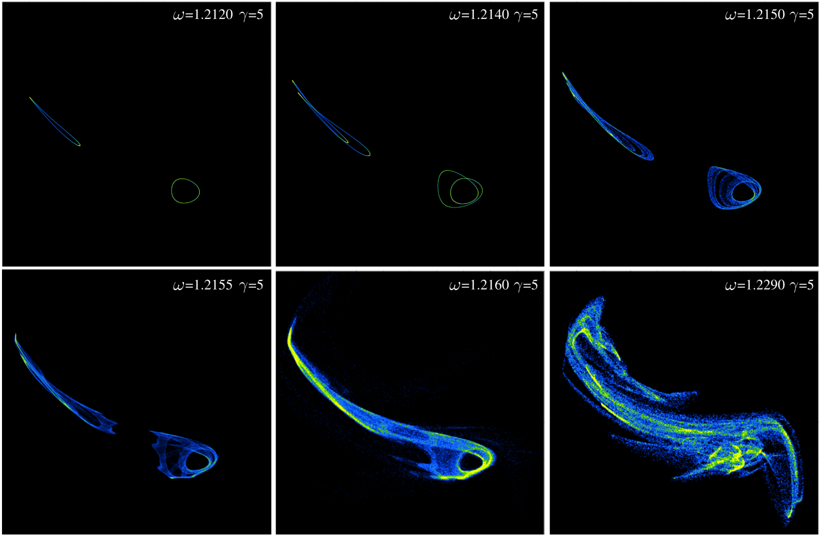
# Transition to chaos: mechanism

Projection of 4D phase portraits on 2D



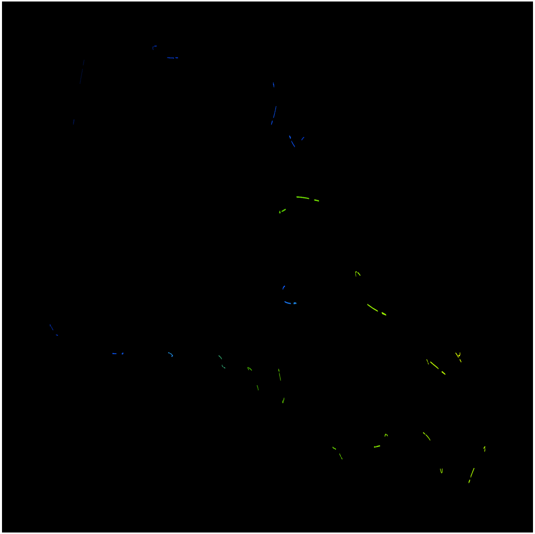

[8] Natsiavas (1993), J Sound Vib 165

# Transition to chaos: example



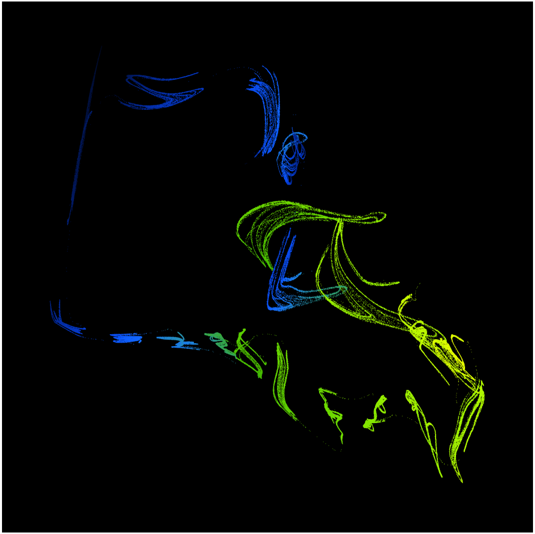

Vizualise in 4D

# Transition to chaos: example 2

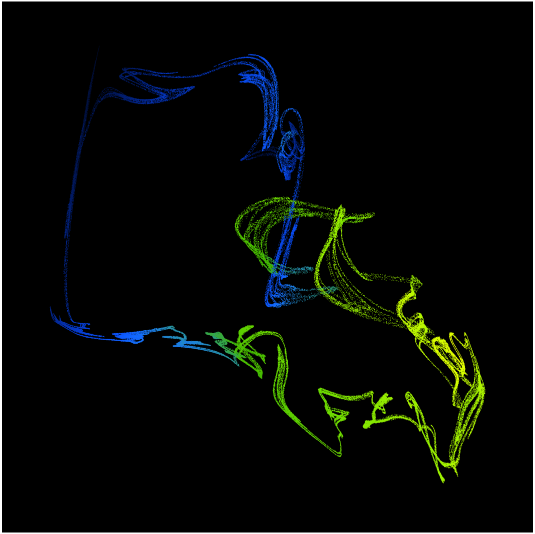


# Transition to chaos: example 2

# Transition to chaos: example 2

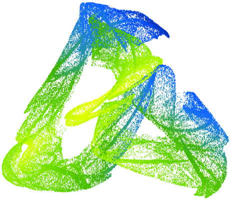


# Transition to chaos: example 2

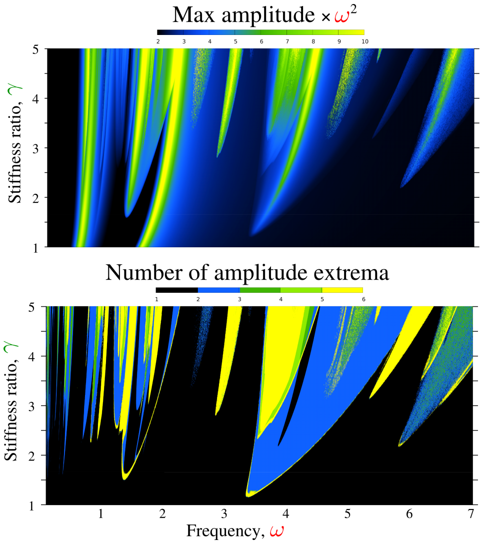


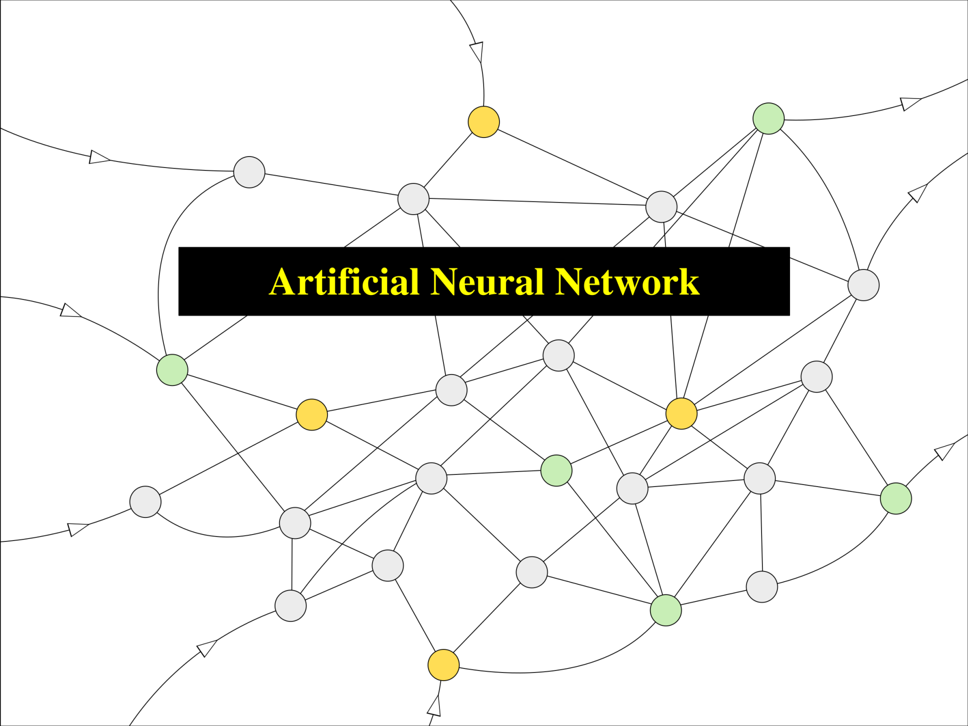
# Strange attractors in 4D

Examples of 4D strange attractor projected on 3D $(x_1, \dot{x}_1, x_2, \dot{x}_2)$, the color represents the 4th dimension

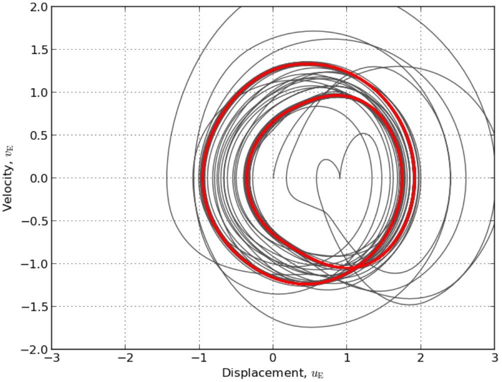


# Lighter damping



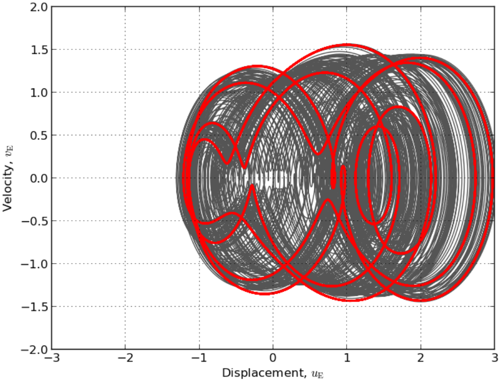

Artificial Neural Network

# Motivation



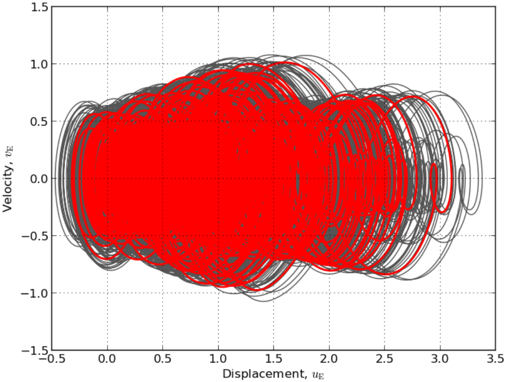

Sometimes finding the limit cycle is fast

# Motivation




Sometimes the stabilisation takes longer time:
simulation of thousands load cycles is needed

# Motivation




Sometimes there's no limit cycle: chaotic regime
To be sure that it's chaotic, millions(!) of cycles should be simulated

# Motivation

Could we train an Artificial Neural Network (ANN) to
predict
the regime of oscillations?

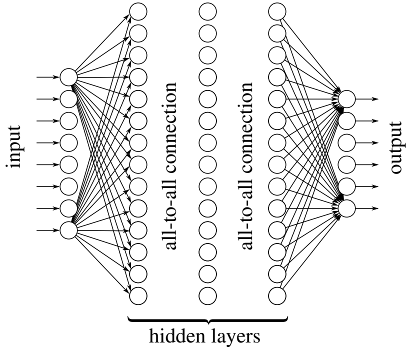
# Objectives & means

**Objectives:**

1. Forecast dynamics without full information about the system
2. Forecast vibration regime (simple, multi-period, quasiperiodic, chaotic)

**Means:**

1. Multilayer Perceptron with ReLU (Rectified Linear Unit)




- Layer to layer formula

$$X^{i+1} = f(W^{i,i+1}X^i + B^i)$$

$W^{i,i+1}$ matrix of weights,
$B^i$ vector of biases.

- ReLU

$$f(x) = \begin{cases} 0, & \text{if } x < 0; \\ x, & \text{if } x \geq 0. \end{cases}$$

- Least mean squared error, backpropagation

# Architectures

- **Input/output:**
  - Independent ANN for velocity and displacement

    `input:` $\{X^{k+i}\}, i \in [1,100]$
    `output:` $\{X^{k+100+j}\}, j \in [1,50]$

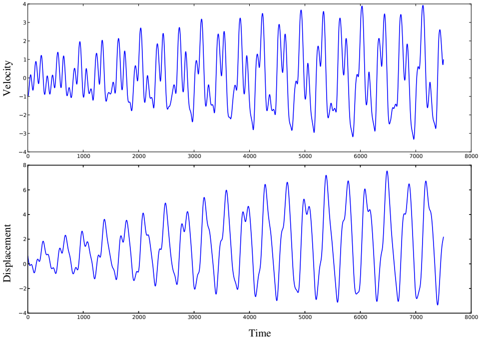
  - Single ANN for both velocity and displacement

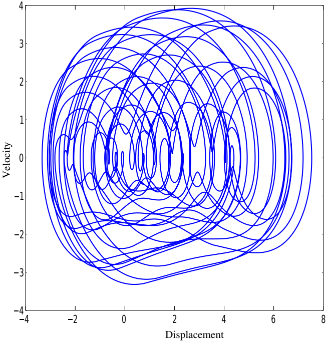
    `input:` $\{X^{k+i}, V^{k+i}\}, i \in [1,100]$
    `output:` $\{X^{k+100+j}, V^{k+100+i}\}, j \in [1,50]$

- **Architectures:**
  Inp time points → Hidden Layer → Hidden Layer → ⋯ → Out time points
  - One hidden layer: 2×100 → 1000 → 2×50
  - Two hidden layers: 2×100 → 500 → 200 → 2×50
  - Three hidden layers: $100 \to \underbrace{100 \to 75 \to 50}_{\text{Hidden layers}} \to 50$
  - Three hidden layers: $2 \times 100 \to \underbrace{200 \to 150 \to 100}_{\text{Hidden layers}} \to 2 \times 50$
  - etc ...

# Objective 1: forecasting

## Training



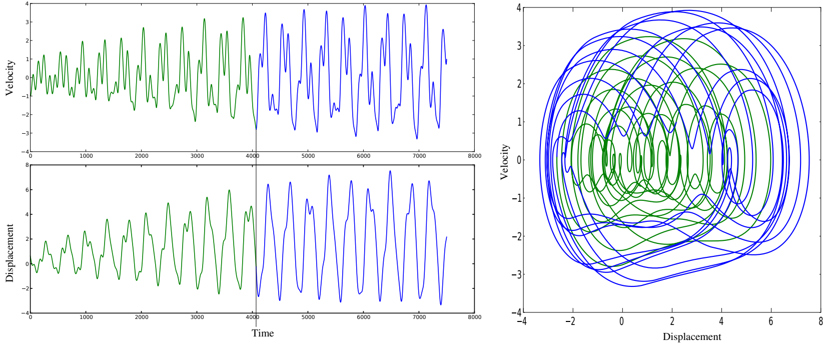




## Forecasting

- Within training interval
- Outside training interval

# Objective 1: forecasting

**Training**



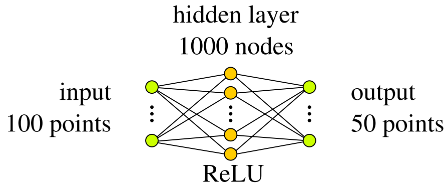

In total 100 cycles simulated: 50 cycles for training, 50 for verification

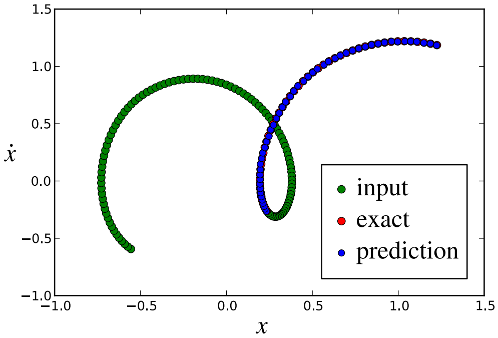
**Forecasting**

- Within training interval
- Outside training interval

# Results I

**Single layer:** complexity $200 \times 1000 + 1000 \times 100 = 3 \cdot 10^5$

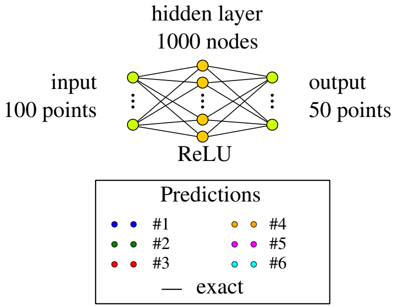


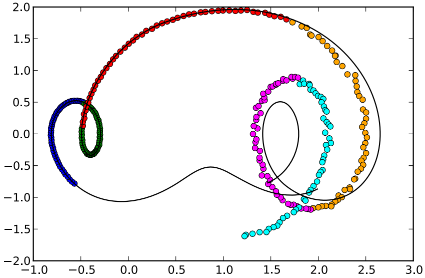




Single prediction

# Results I

**Single layer:** complexity $200 \times 1000 + 1000 \times 100 = 3 \cdot 10^5$

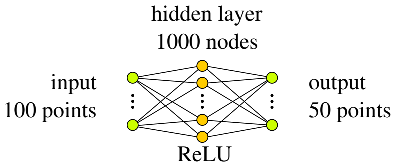


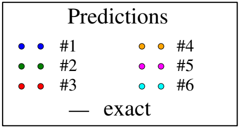



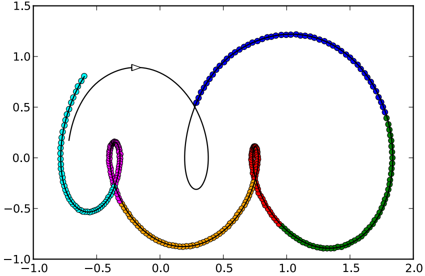

Six predictions

# Results I

**Single layer:** complexity $200 \times 1000 + 1000 \times 100 = 3 \cdot 10^5$



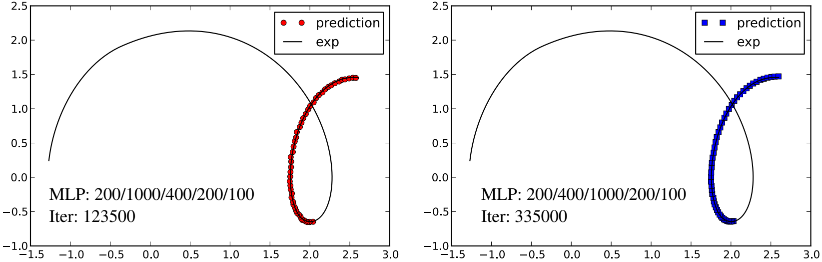




Sinx predictions (reinforced convergence tolerance)

# Results II: inside vs outside training domain

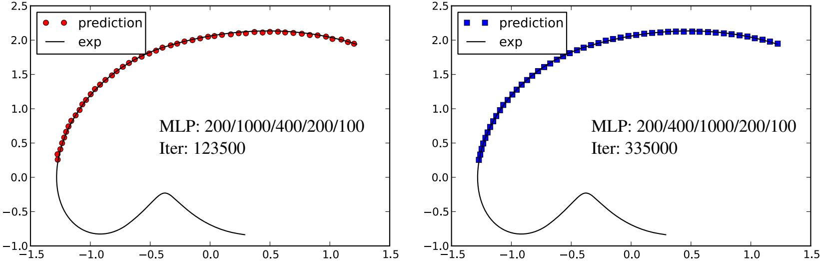
**Three layers:** complexity $7 \cdot 10^5$




Within training domain

# Results II: inside vs outside training domain

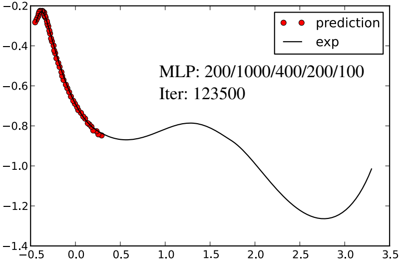
**Three layers:** complexity $7 \cdot 10^5$

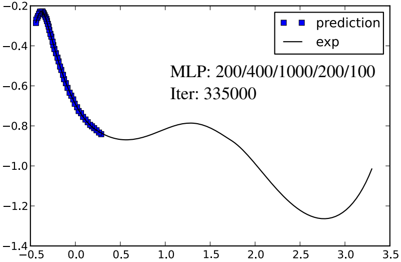



Within training domain

# Results II: inside vs outside training domain

**Three layers:** complexity $7 \cdot 10^5$

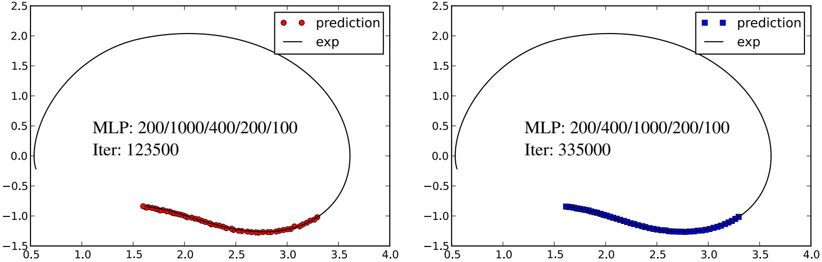



Within training domain

# Results II: inside vs outside training domain

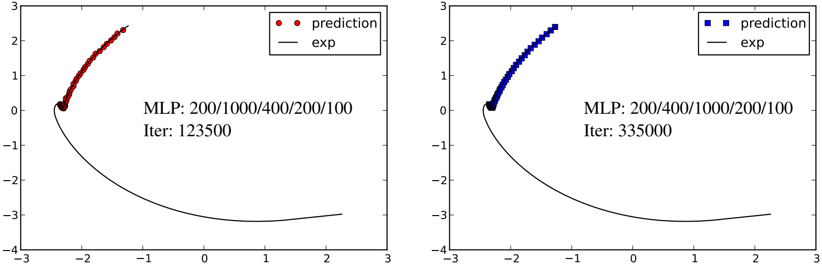
**Three layers:** complexity $7 \cdot 10^5$




Within training domain

# Results II: inside vs outside training domain

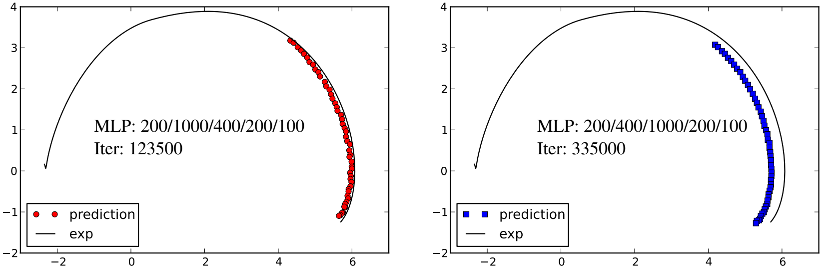
**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

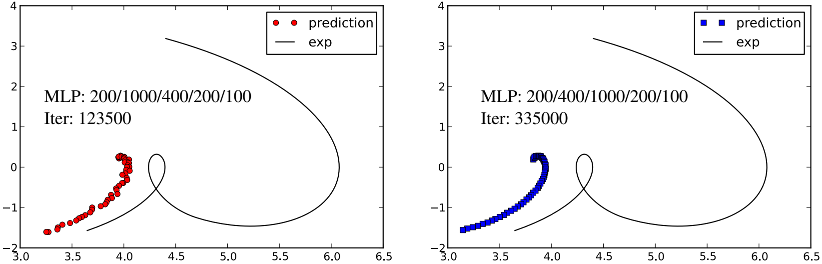
# Results II: inside vs outside training domain

**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

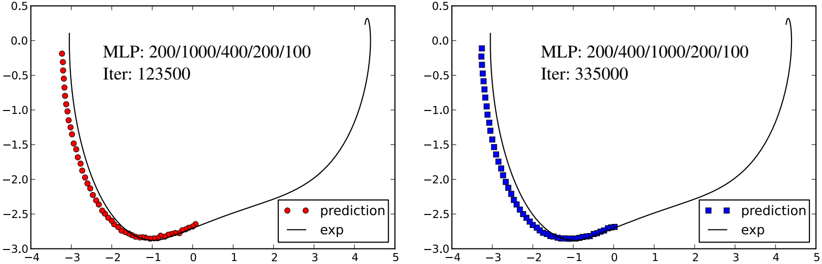
**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

# Results II: inside vs outside training domain

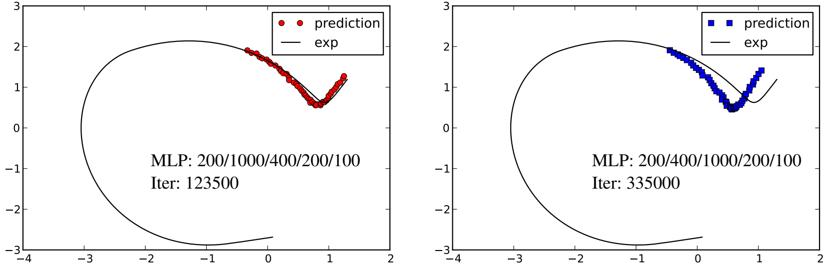
**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

# Results II: inside vs outside training domain

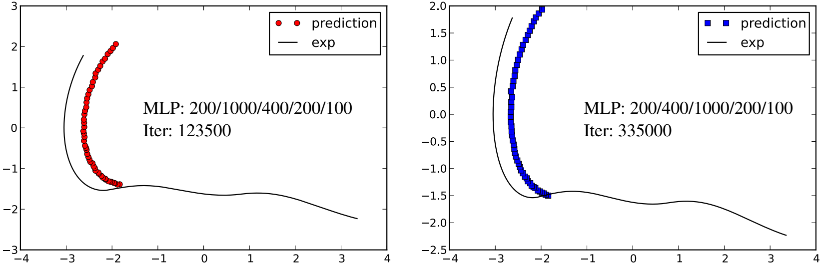
**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

# Results II: inside vs outside training domain

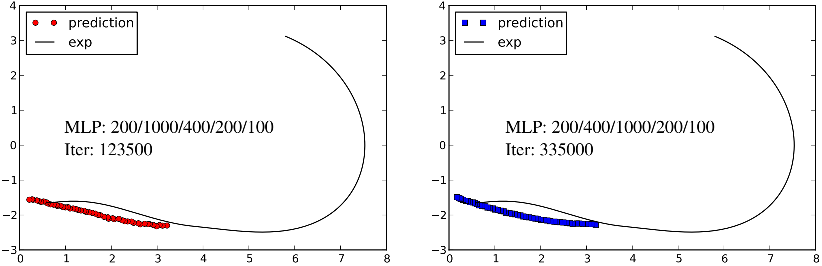
**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

# Results II: inside vs outside training domain

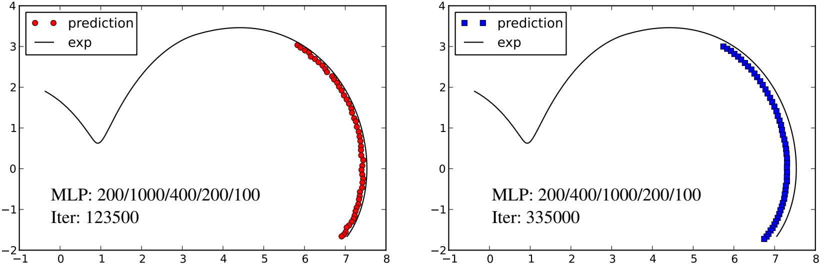
**Three layers:** complexity $7 \cdot 10^5$




Outside training domain

# Results II: inside vs outside training domain

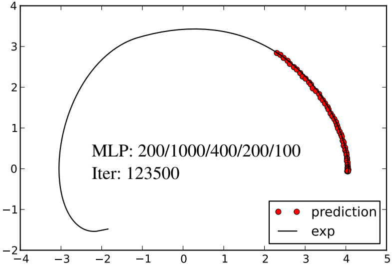
**Three layers:** complexity $7 \cdot 10^5$

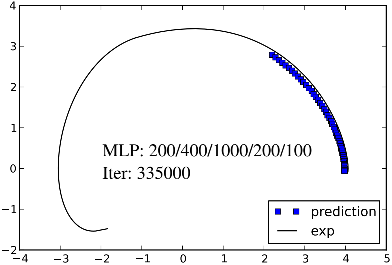



Outside training domain

# Results II: inside vs outside training domain

**Three layers:** complexity $7 \cdot 10^5$

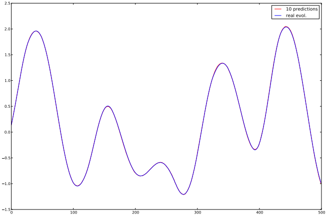


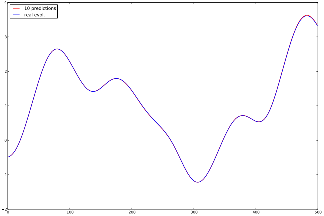




Outside training domain

# Results III: Simplified architecture

**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$



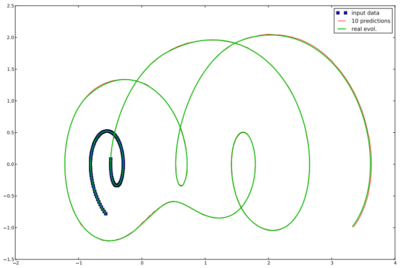




Velocity Displacement

10 predictions within training domain

# Results III: Simplified architecture

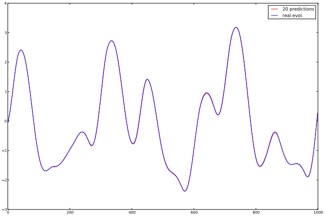
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$

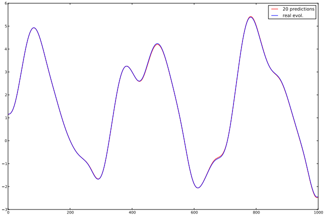



10 predictions within training domain

# Results III: Simplified architecture

**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$



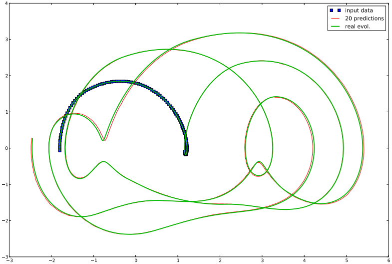




Velocity Displacement

20 predictions within training domain

# Results III: Simplified architecture

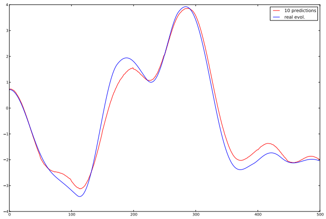
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$

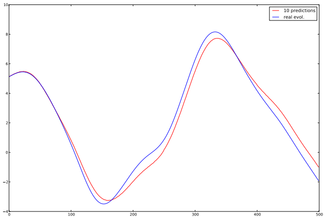



10 predictions within training domain

# Results III: Simplified architecture

**Three layers:** $200 \rightarrow 100 \rightarrow 100 \rightarrow 100 \rightarrow 100$, complexity $5 \cdot 10^4$



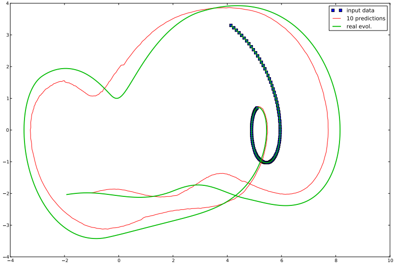




Velocity Displacement

10 predictions **outside** training domain

# Results III: Simplified architecture

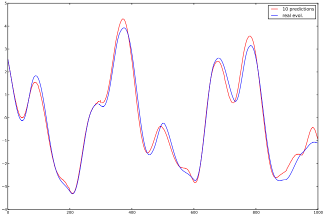
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$

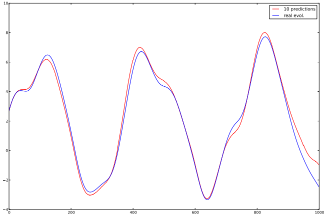



10 predictions **outside** training domain

# Results III: Simplified architecture

**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$



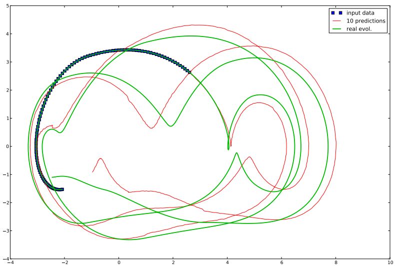




Velocity　　　　　　　　　　Displacement

20 predictions **outside** training domain

# Results III: Simplified architecture

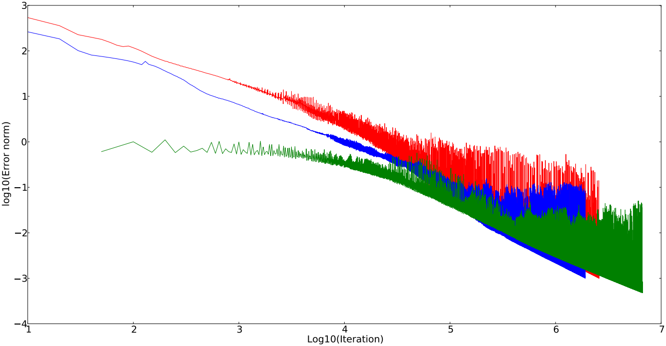
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$




20 predictions **outside** training domain

# Results III: Simplified architecture

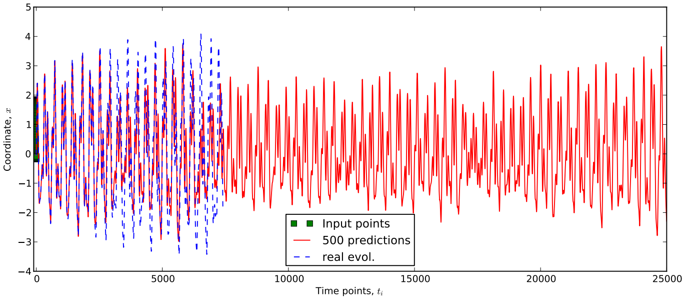
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$




Convergence for different learning rates

# Results IV: go futher in prediction

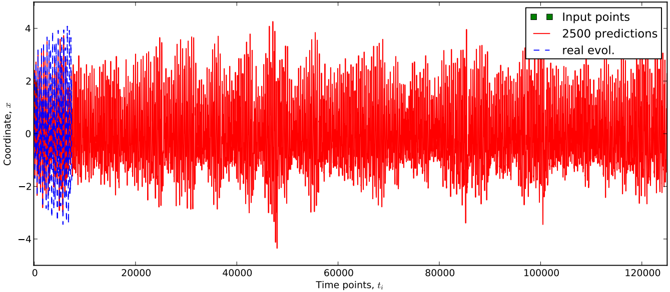
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$




500 predictions from the inside of the training domain

# Results IV: go futher in prediction

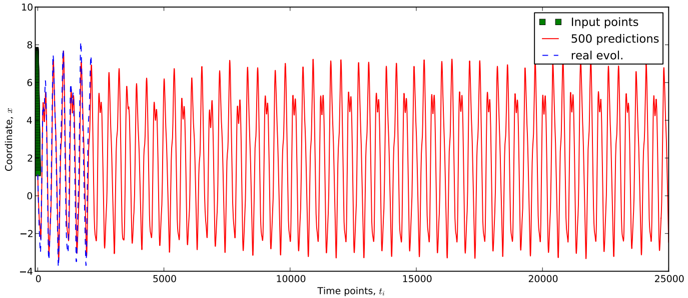
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$




2500 predictions from the inside of the training domain

# Results IV: go futher in prediction

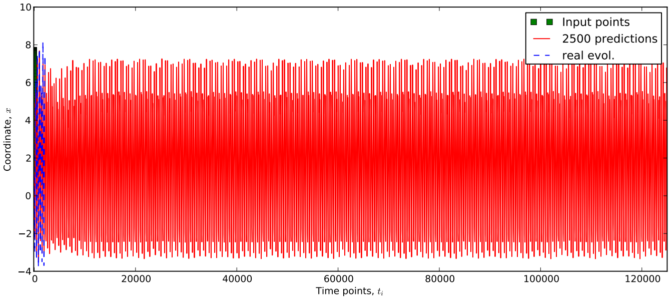
**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$




500 predictions from the inside of the training domain

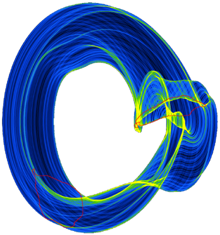
# Results IV: go futher in prediction

**Three layers:** $200 \to 100 \to 100 \to 100 \to 100$, complexity $5 \cdot 10^4$




2500 predictions from the inside of the training domain

# Conclusions & perspectives



# Conclusions and perspectives

**First results**

- To some extend we can forecast complex dynamics accurately within the training domain, less accurately outside it.
- However, we trained our ANN only for a single frequency

**Perspectives**

- Predicting dynamics for arbitrary frequency seems too difficult, hm? Too much training data and time is needed.
- Classification of regimes for arbitrary frequency seems quite difficult

# Conclusions and perspectives

**First results**

- To some extend we can forecast complex dynamics accurately within the training domain, less accurately outside it.
- However, we trained our ANN only for a single frequency

**Perspectives**

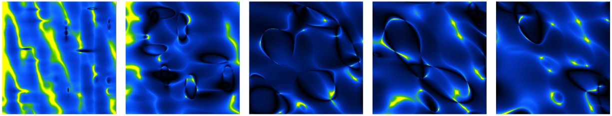
- Predicting dynamics for arbitrary frequency seems too difficult, hm? Too much training data and time is needed.
- Classification of regimes for arbitrary frequency seems quite difficult

# Thank you for your attention!

# Merci de votre attention !

<vladislav.yastrebov@mines-paristech.fr>
www.yastrebov.fr

# Computational approach



# Computational strategies

## Integration

1. Direct integration (explicit/implicit)
   - Störmer-Verlet integration (lacks precision)
2. Semi-analytical integration
   - Newton method to solve homogeneous problem
   - Newton method to identify switch between regimes
   - Adapt time step to match $2\pi/\omega$

- *A few regimes with constant coefficients*
- *Thus they are integrable semi-analytically*

## Results

- Random assignment of initial conditions
- Periodicity: check history of Poincaré points (depth $2^n$)
- Particular treatment of chaotic regimes