Institute of Parallel and Distributed High-Performance Systems (IPVR)
University of Stuttgart, Germany

Remarks to simulation and investigation of hybrid systems

Viktor Avrutin

E-mail: vravrout@informatik.uni-stuttgart.de

Contents

- Modeling: from hybrid automata to classical dynamical systems.
  - hybrid automata
  - hybrid equations
  - connections and global context
- Investigation methods for hybrid systems
- Simulation: (not only) hybrid systems
  - **AnT 4.667** software package

HYBRID APPROACH

Main idea:

"A hybrid system is a dynamical system whose behavior exhibits both discrete and continuous change."

Examples for application areas:

- mechanical systems
- electrical systems
- biological systems
- chemical systems
- software-controlled systems
- . . .

Question:

What are the consequences for the following tasks:

modeling

$\Downarrow$

simulation

$\Downarrow$

analysis

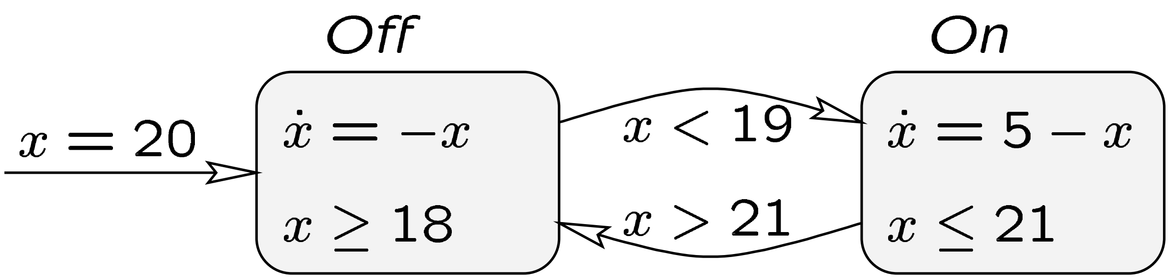
?

Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

Main components:

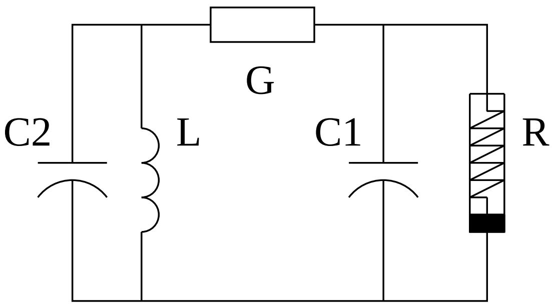
- Finite set of real–numbered variables $\vec{X}$.
- Finite graph (V, E), with V – control modes, E – control switches.
- Transition functions $\dot{\vec{X}} = \vec{f}(\vec{X})$ for each control mode.
- Invariance conditions for each control mode.
- Jump conditions for each control switch.
- Finite set of events Σ

Standard example: Thermostat automaton



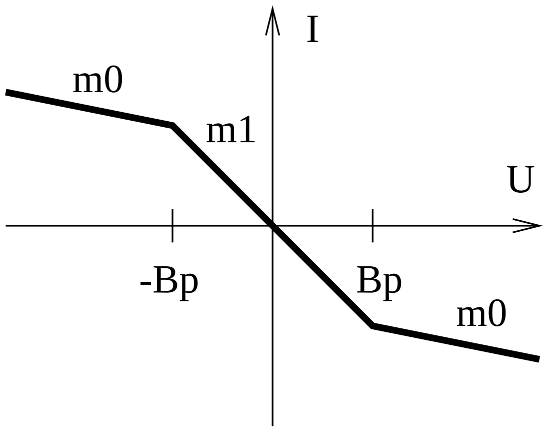
EXAMPLE: STANDARD CHUA SYSTEM

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)



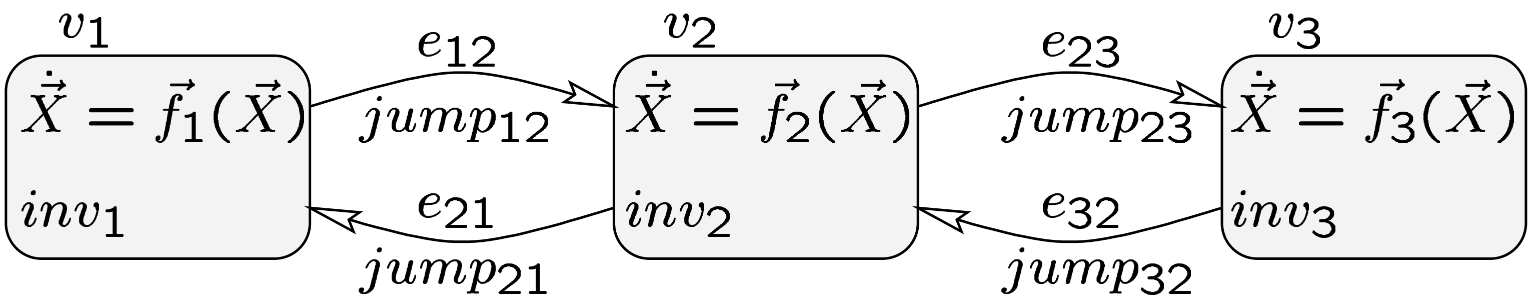
variables: U_{C_1}, U_{C_2} and I_L

$$
\begin{aligned}
C_1 \dot{U}_{C_1} &= G(U_{C_2} - U_{C_1}) - g(U_{C_1}) \\
C_2 \dot{U}_{C_2} &= G(U_{C_1} - U_{C_2}) + I_L \\
L \dot{I}_L &= -U_{C_2}
\end{aligned}
$$



$$
g(U_R) = \begin{cases} m_0 U_R + B_p(m_1 - m_0) & \text{if} \quad U_R > B_p \\ m_1 U_R & \text{if} \quad |U_R| \leq B_p \\ m_0 U_R - B_p(m_1 - m_0) & \text{if} \quad U_R < -B_p \end{cases}
$$

HYBRID AUTOMATON FOR STANDARD CHUA SYSTEM



$$\begin{aligned}
\vec{X} &= (x, y, z)^T \in \mathbb{R}^3 \\
V &= \{v_1, v_2, v_3\} \\
E &= \{e_{12}, e_{21}, e_{23}, e_{32}\} \\
\Sigma &= \{\sigma_{12}, \sigma_{23}\} \\
Inv &= \{inv_1 : x < -1, \\
&\quad\; inv_2 : -1 \leq x \leq 1 \\
&\quad\; inv_3 : x > 1\} \\
Jump &= \{jump_{12}, jump_{21}, \\
&\quad\; jump_{23}, jump_{32}\}
\end{aligned}$$

$$\vec{f}_1 = \begin{pmatrix} \alpha(y - (b+1)x - a + b) \\ x - y + z \\ -\beta y \end{pmatrix}$$

$$\vec{f}_2 = \begin{pmatrix} \alpha(y - (a+1)x) \\ x - y + z \\ -\beta y \end{pmatrix}$$

$$\vec{f}_3 = \begin{pmatrix} \alpha(y - (b+1)x + a - b) \\ x - y + z \\ -\beta y \end{pmatrix}$$

Note: consistence problems for Inv and $Jump$!

From viewpoint of dynamical systems:

- :-) well–structured modeling, detailed model validation

- :-| often complex (not generic) simulation

- :-(not well-elaborated investigation methods: typical question – reachability problem, typical system class – linear hybrid automata, where $\dot{\vec{X}} = const$ for all control modes

Note: transition functions for the continuous state $\vec{X}$ must be not necessary given by ODEs. Others system types (maps, DDEs) can be used as well!

PROBLEMS OF HYBRID AUTOMATA

Problems occur, if we have to model a system, where more as one components show dynamics with several modes of operation.

Example: standard producer/consumer system.

State of the producer: x.
Modes of operation: m_1^p (producing) and m_2^p (waiting).

State of the consumer: y.
Modes of operation: m_1^c (consuming) and m_2^c (waiting).

State of the system $(x, y)^T$. Modes of operation:
$m_1 = [m_1^p, m_1^c]$, $m_2 = [m_2^p, m_1^c]$, $m_3 = [m_1^p, m_2^c]$, $m_4 = [m_2^p, m_2^c]$

Note: This representation has advantages (we see, that m_4 corresponds to a deadlock), but the complexity of the model grows $\sim (n_1 \cdot n_2 \cdot \ldots)$ instead of $\sim (n_1 + n_2 + \ldots)$.

Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

Main components:

- Finite set of real-numbered variables $\vec{X}$.
- Finite set of discrete-valued variables $\vec{m}$.
- Transition functions

$$
\begin{aligned}
\dot{\vec{X}}(t) &= \vec{f}(\vec{X}(t), \vec{m}(t)) \\
\vec{m}(t^{+}) &= \vec{\phi}(\vec{X}(t), \vec{m}(t))
\end{aligned}
$$

From viewpoint of dynamical systems we have here a set conststing of an ODE an a map ϕ, continuous in time, but discrete–valued.

Note: if needed, the model can be extended by external input, external events, etc., but this is not a generic case.

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)

Hybrid state vector $\vec{X} = (x, y, z, m)^T$, with continuous variables $x, y, z \in \mathbb{R}$ and a discrete variable $m \in \{m_1, m_2, m_3\}$

$$\dot{x} = \begin{cases} \alpha(y - x - bx - a + b) & \text{if} \quad m = m_1 \\ \alpha(y - x - ax & \text{if} \quad m = m_2 \\ \alpha(y - x - bx + a - b & \text{if} \quad m = m_3 \end{cases}$$

$$\dot{y} = x - y + z$$

$$\dot{z} = -\beta y$$

$$m = \begin{cases} m_1 & \text{if} \quad x > 1 \\ m_2 & \text{if} \quad -1 \leq x \leq 1 \\ m_3 & \text{if} \quad x < -1 \end{cases}$$

The values m_1, m_2, m_3 of the variable m correspond to control modes v_1, v_2, v_3 of the hybrid automaton presented above.

Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

From viewpoint of dynamical systems:

- :-(quite complex modeling,
- :-| more generic simulation,
- :-) most investigation methods for classical dynamical systems are applicable.

SUMMARY TO HYBRID SYSTEMS

hybrid automata	hybrid equations with n discrete variables
⇓	⇓

hybrid equations with 1 discrete variable

⇕

dynamical systems with piecewise defined system function

- Hybrid automata are more suitable for modelling and model validation
- Hybrid equations are more suitable for simulation and investigation

Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

Piecewise–defined vector flow, specific for hybrid systems, can have negative consequences, no consequences, or positive consequences concerning the calculation of several quantitative measures of the dynamics occurring in these systems.

- negative consequences:
  Lyapunov exponents, ...
- no consequences:
  power spectra, fractal dimensions, ...
- positive consequences:
  symbolic–based measures (entropies of symbolic sequences, etc.), methods using Poincaré sections, ...

Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

Approach: Lyapunov exponents will be calculated on the basis of the continuous part of the state space only.

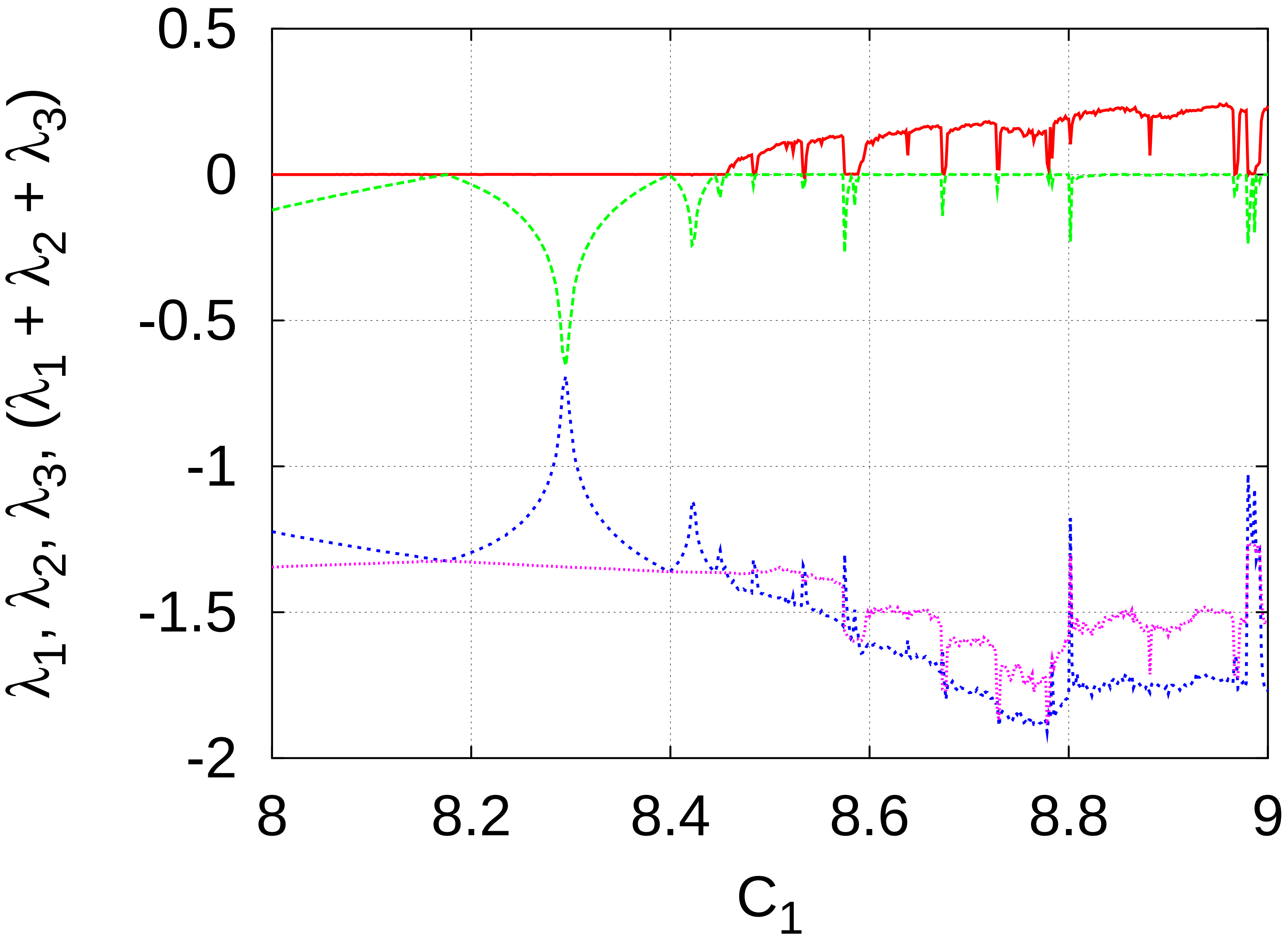
- *("theoretical point of view")*
  Lyapunov exponents are (until now) not well defined for hybrid systems, bacause the exact mathemathical definition of these values requires a differentiable vector flow, which is obviously not given by hybrid dynamical systems.

- *("pragmatical point of view")*
  Some numerical approaches for calculation of Lyapunov exponents produce for hybrid systems values, which are **not** Lyapunov exponents, but seem to be very like with it.

Example: Approach of Wolf *et. al* approximate Lyapunov exponents with divergence rates, averaged over time.

"LYPUNOV EXPONENTS" FOR STANDARD CHUA SYSTEM

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)



Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

- Applying of symbolic dynamics requires two steps:

  - generation of symbolic sequences

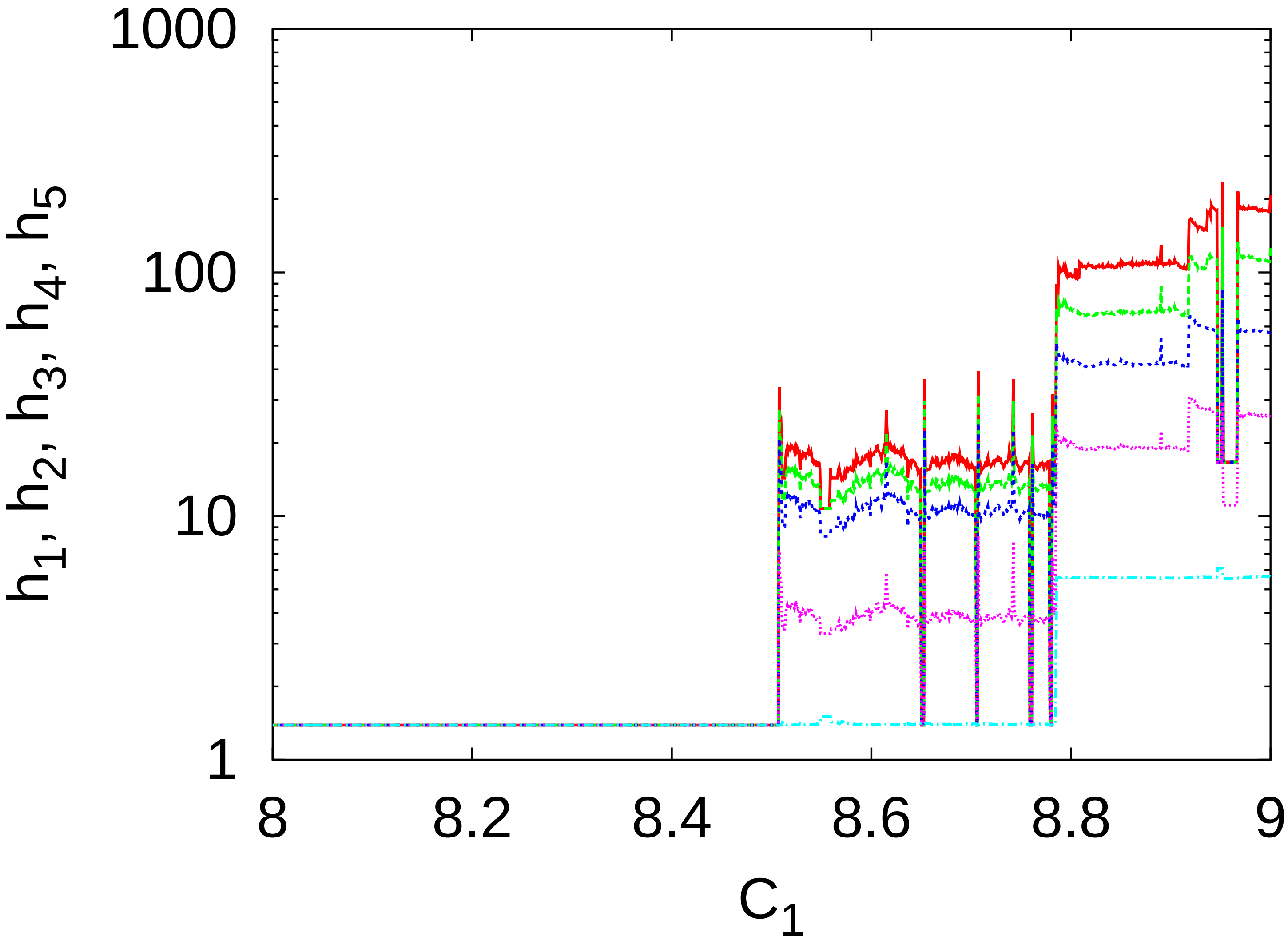
    There exists some generic approaches, such as $(\mathcal{L}/\mathcal{R})-$ and $(+/-)$ symbolic dynamics, as well as several specific for individual dynamic systems partitions of the state space.

  - evaluation of symbolic sequences.

    There exists some quantitative measures, which can be calculated from a symbolic sequence, such as its entropies, etc.

- For orbits of hybrid systems symbolic sequences can be calculated from the discrete-valued part of the orbit. Approach: control modes define the symbols of the symbolic dynamics.

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)



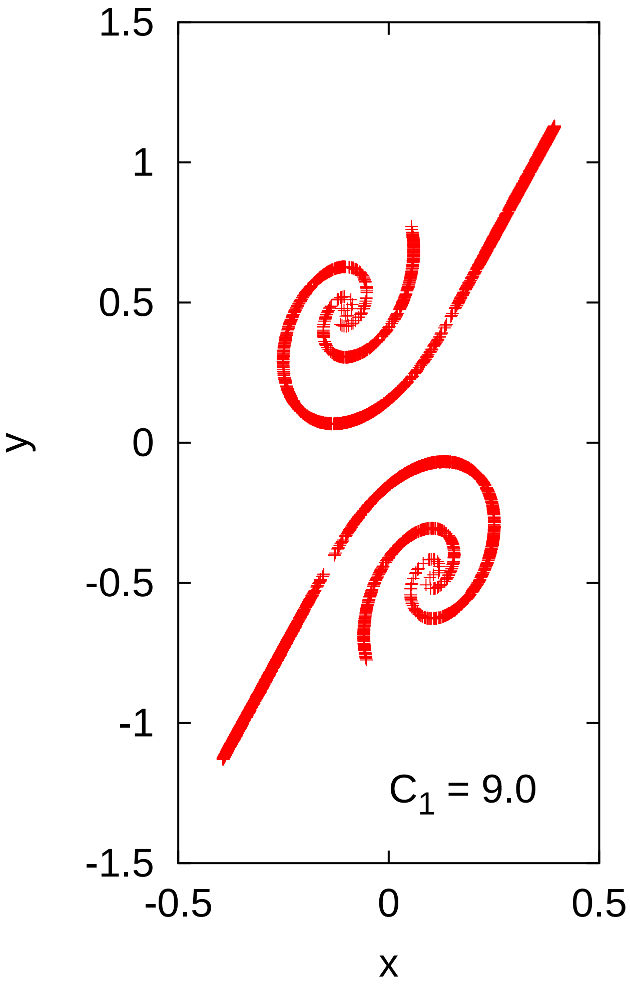
Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

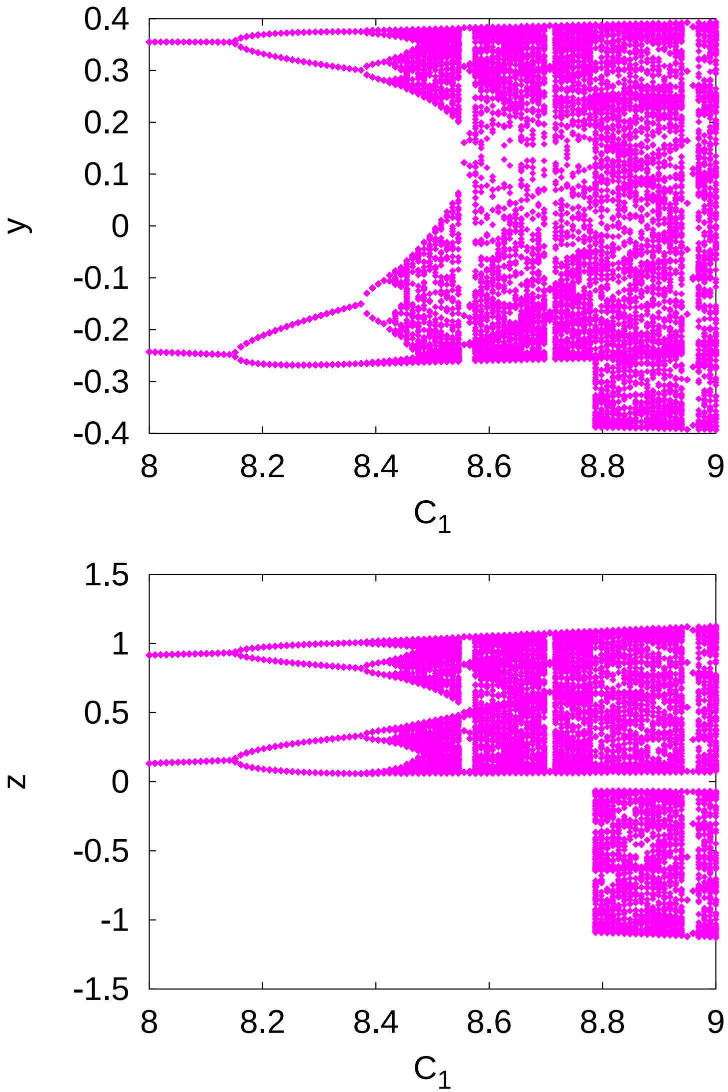
- **Standard Poincaré sections:**
  Cross-section points of the orbit with a (hyper)–plane in the state space.

- **Generalized Poincaré sections:**
  Points in the state space, where some specifec condition concerning the orbit is fulfilled.

- **Generic Poincaré section for hybrid systems:**
  Points in the continuous sub–space of the hybrid state space, where the orbit changes its state in the discrete sub–space of the state space. (Change the control mode of the hybrid automaton, changing the value of the discrete state vector, etc).

POINCARÉ SECTIONS FOR STANDARD CHUA SYSTEM





what is **AnT 4.667** ?

AnT 4.667 – a simulation and analysis tool for dynamical systems

- main application areas:
  science and education

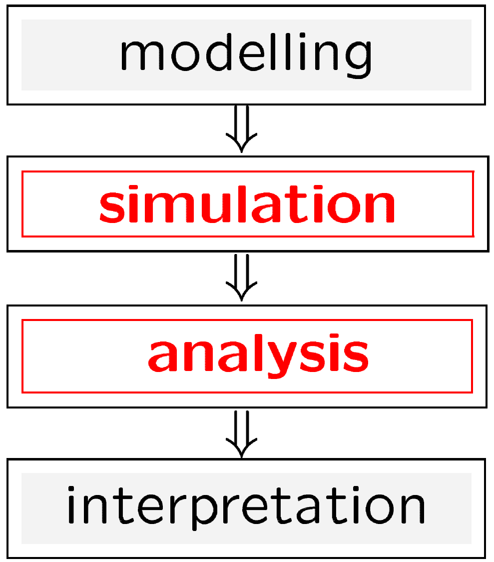
- some properties:

  - support of several classes of dynamical systems

  - several investigation methods

  - scan-capabilities

  - open software architecture



Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)

AnT 4.667 is developed by the Non–Linear Dynamics Group of the Department of Image Understanding (Head: Prof. Dr. P. Levi) at the Institute for Parallel and Distributed High-Performance Systems of the University of Stuttgart.

Members of the group:

- Viktor Avrutin,
- Robert Lammert,
- Dr. Michael Schanz,
- Heiko Schäfer,
- Michael Schulze,
- Sascha Riexinger,
- Georg Wackenhut.

History of the project:

1998:	first prototypes (FORTRAN, C)
2000:	**AnT 4.66** (C)
2001:	**AnT 4.667** (C++)

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)

- maps: $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \{\sigma\})$
- recurrent maps: $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \vec{x}_{n-1}, ..., \vec{x}_{n-\tau}, \{\sigma\})$
- ODEs: $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \{\sigma\})$
- DDEs: $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{x}(t-\tau), \{\sigma\})$
- FDEs: $\dot{\vec{x}}(t) = \vec{f}\,[\vec{x}_t, \{\sigma\}]$
  with $\vec{x}_t(\theta) = \vec{x}(t+\theta)$, $\theta \in [-\tau, 0])$
- PDEs: $\frac{\partial}{\partial t}\vec{x}(q,t) = \vec{f}\left(\vec{x}(q,t), \frac{\partial}{\partial q}\vec{x}(q,t), \ldots, \{\sigma\}\right)$
- external data input $\vec{x}_{n+1}$ = next input vector

- CMLs

$$\vec{x}^{(i)}_{n+1} = \vec{f}\left(\vec{x}^{(i-r)}_{n}, \ldots, \vec{x}^{(i)}_{n}, \vec{x}^{(i+r)}_{n}, \{\sigma\}\right)$$

- CODELs

$$\dot{\vec{x}}^{(i)}(t) = \vec{f}\left(\vec{x}^{(i-r)}(t), \ldots, \vec{x}^{(i)}(t), \ldots, \vec{x}^{(i+r)}(t), \{\sigma\}\right)$$

- CDDELs

$$\begin{aligned}\dot{\vec{x}}^{(i)}(t) \quad = \quad & \vec{f}\Big(\vec{x}^{(i-r)}(t), \ldots, \vec{x}^{(i)}(t), \ldots, \vec{x}^{(i+r)}(t) \\ & \vec{x}^{(i-r)}(t-\tau), \ldots, \vec{x}^{(i)}(t-\tau), \ldots, \\ & \vec{x}^{(i+r)}(t-\tau), \{\sigma\}\Big)\end{aligned}$$

- hybrid maps

$$\begin{aligned} \vec{x}_{n+1} &= \vec{f}(\vec{x}_n, \vec{m}_n, \{\sigma\}) \\ \vec{m}_{n+1} &= \vec{\phi}(\vec{x}_n, \vec{m}_n, \{\sigma\}) \end{aligned}$$

- hybrid ODEs

$$\begin{aligned} \dot{\vec{x}}(t) &= \vec{f}(\vec{x}(t), \vec{m}(t), \{\sigma\}) \\ \vec{m}(t^+) &= \vec{\phi}(\vec{x}(t), \vec{m}(t), \{\sigma\}) \end{aligned}$$

- hybrid DDEs

$$\begin{aligned} \dot{\vec{x}}(t) &= \vec{f}(\vec{x}(t), \vec{x}(t-\tau), \vec{m}(t), \{\sigma\}) \\ \vec{m}(t^+) &= \vec{\phi}(\vec{x}(t), \vec{m}(t), \{\sigma\}) \end{aligned}$$

Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)

- stochastical systems, based on:
  - . . . maps: $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \{\sigma\}) + \vec{\eta}$
  - . . . ODEs: $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \{\sigma\}) + \vec{\eta}$
  - . . . DDEs: $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{x}(t-\tau), \{\sigma\}) + \vec{\eta}$

- averaged stochastical systems (for instance, maps)

$$\vec{x}_{n+1} = \frac{1}{N} \sum_{i=1}^{N} \left(\vec{f}(\vec{x}_n^{(i)}, \{\sigma\}) + \vec{\eta}^{(i)} \right)$$

Note: From simulation view point – a class of dynamical systems. From dynamical system view point – a investigation method. We call it a meta dynamical system

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)

- **Poincaré maps**
  - using a fixed plane
  - using a parameter-depending plane
  - using generic state space partition for hybrid systems
  - user defined

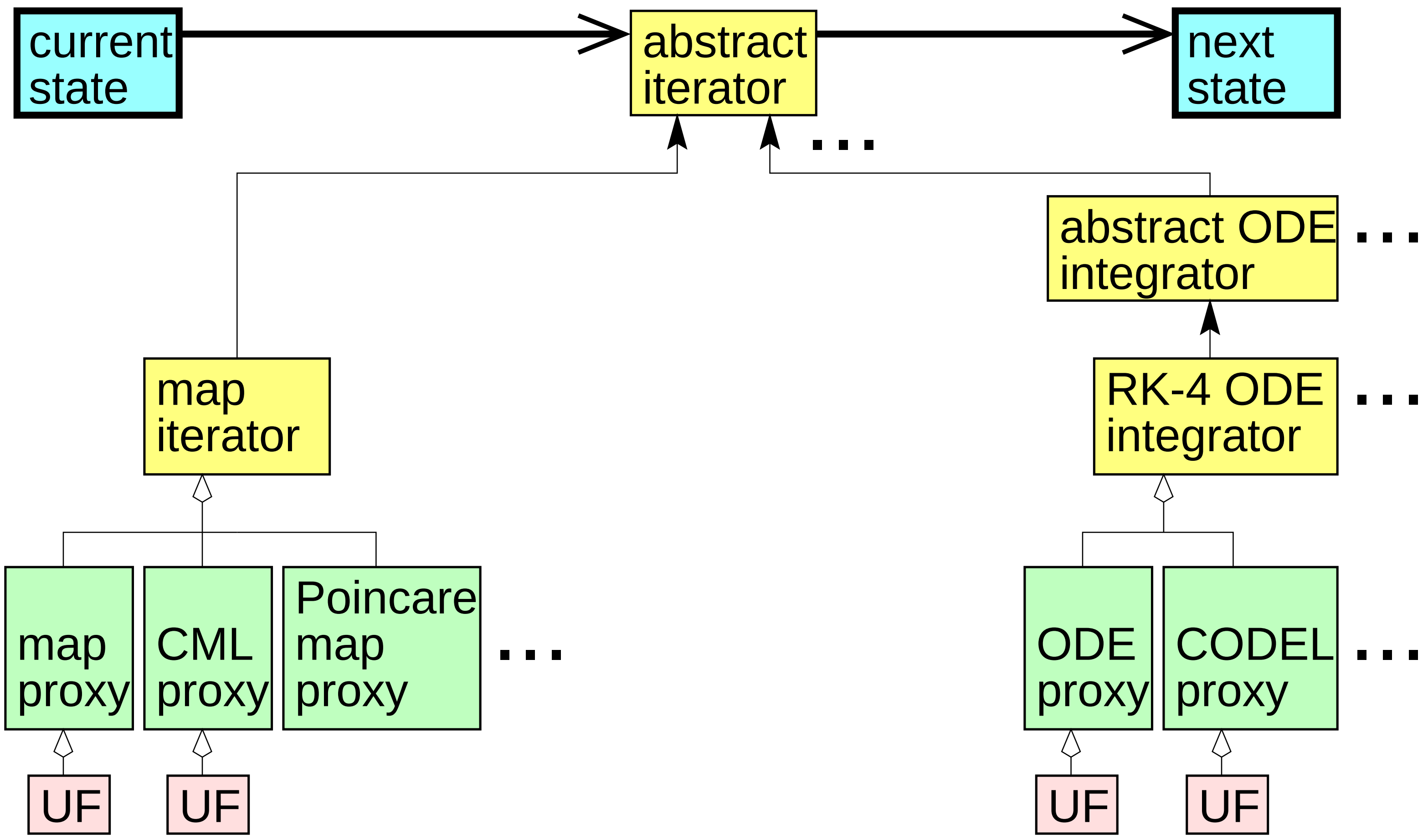
INVESTIGATION METHODS

- Calculation of Lyapunov exponents
- Spectral analysis
- Principal component analysis
- Period analysis
- Symbolic sequence analysis
- Computation of metric entropy and fractal dimensions
- General trajectory evaluations
- ...

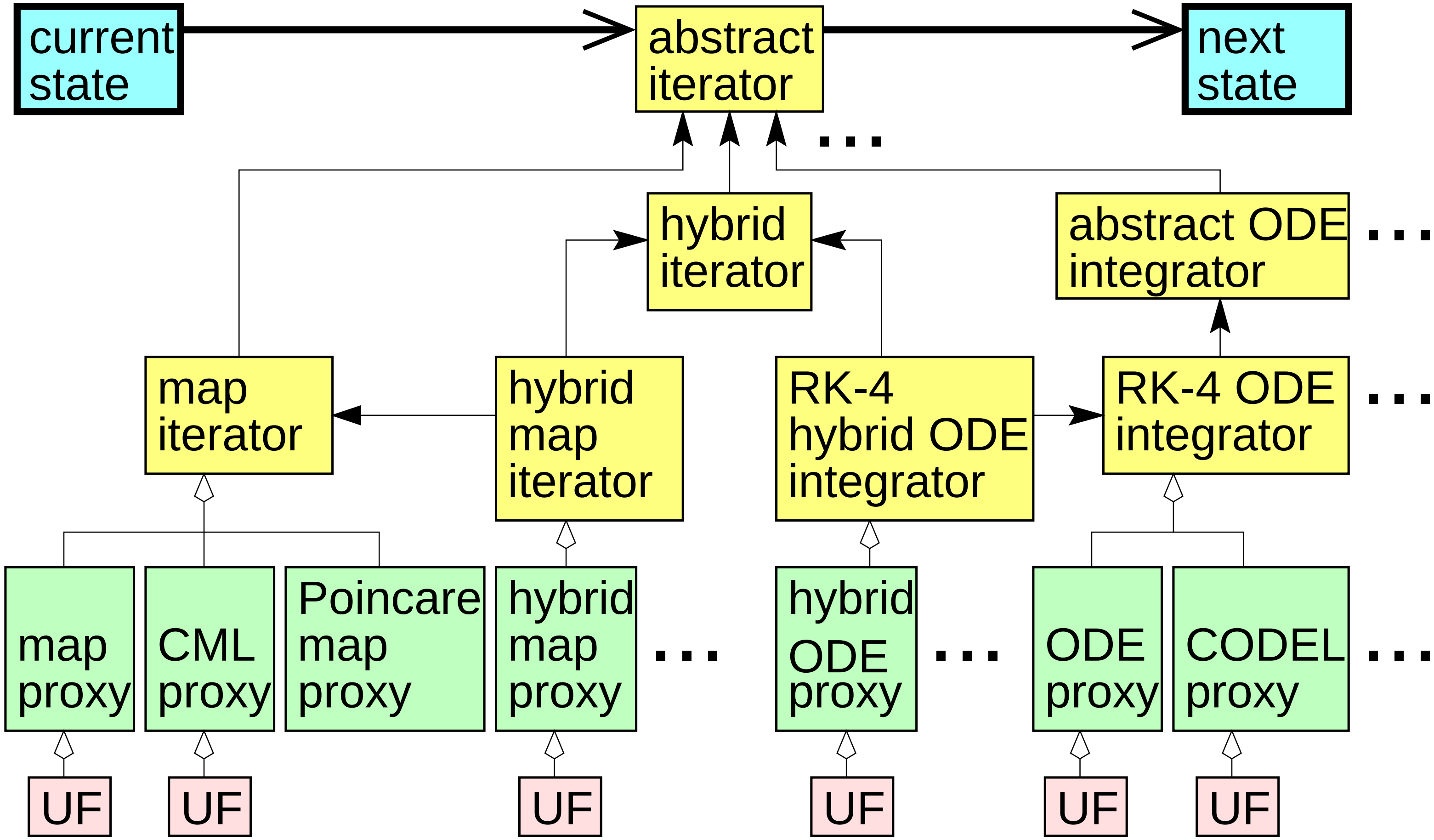
TECHNICAL DETAILS

- very abstract concepts:
  - abstract state
  - abstract transition, transition sequence
  - pre/post state machine, cyclic state machine
- abstract concepts:
  - iterator
  - iter machine, scan machine
- applied concepts:
  - dynamical system data
  - system function proxy
  - simulator

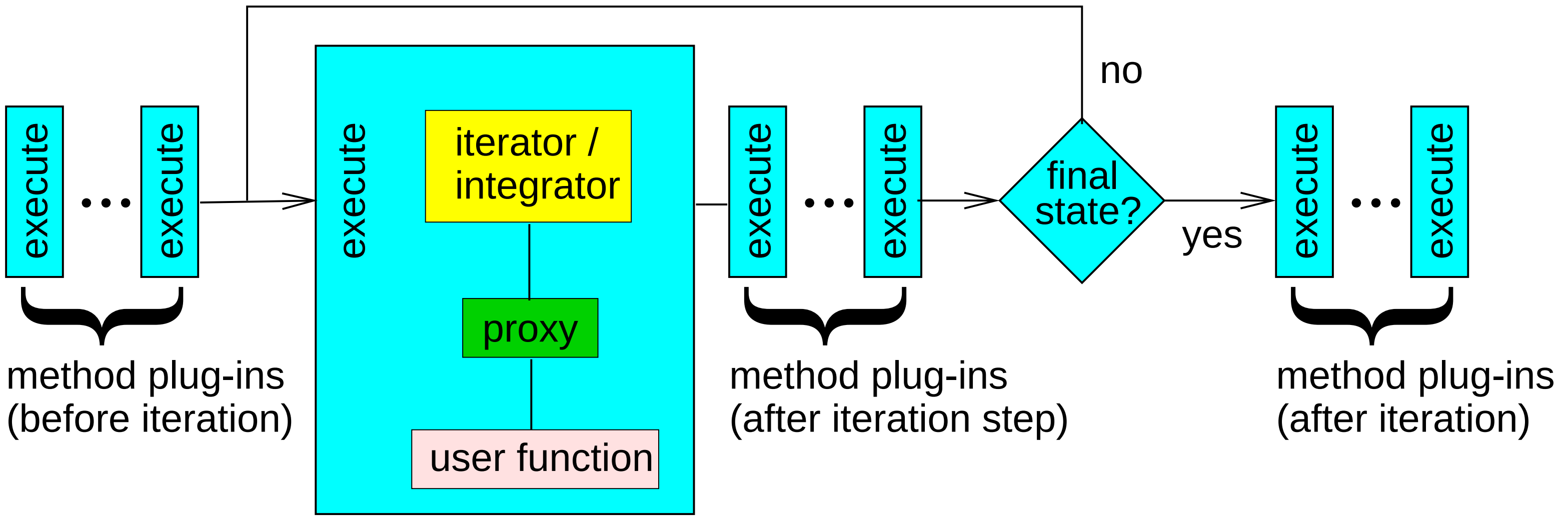
GENERAL ITERATOR



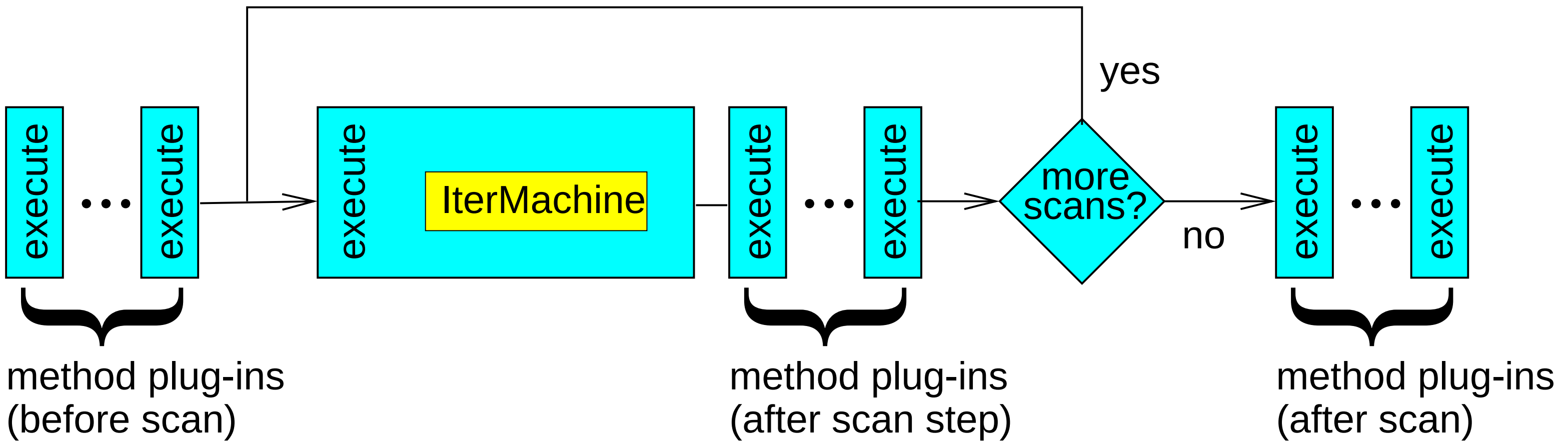
GENERAL ITERATOR



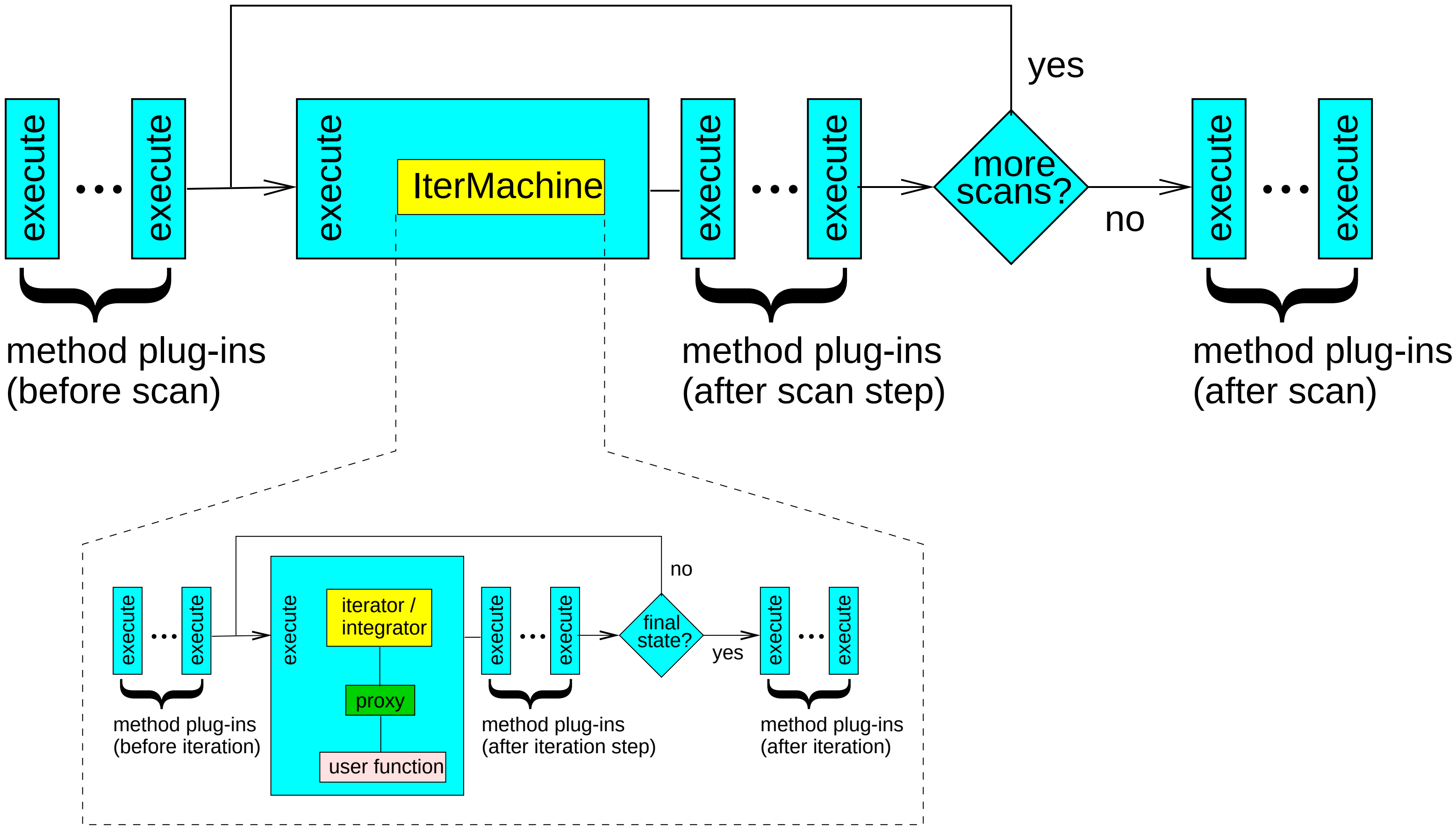
ITER MACHINE



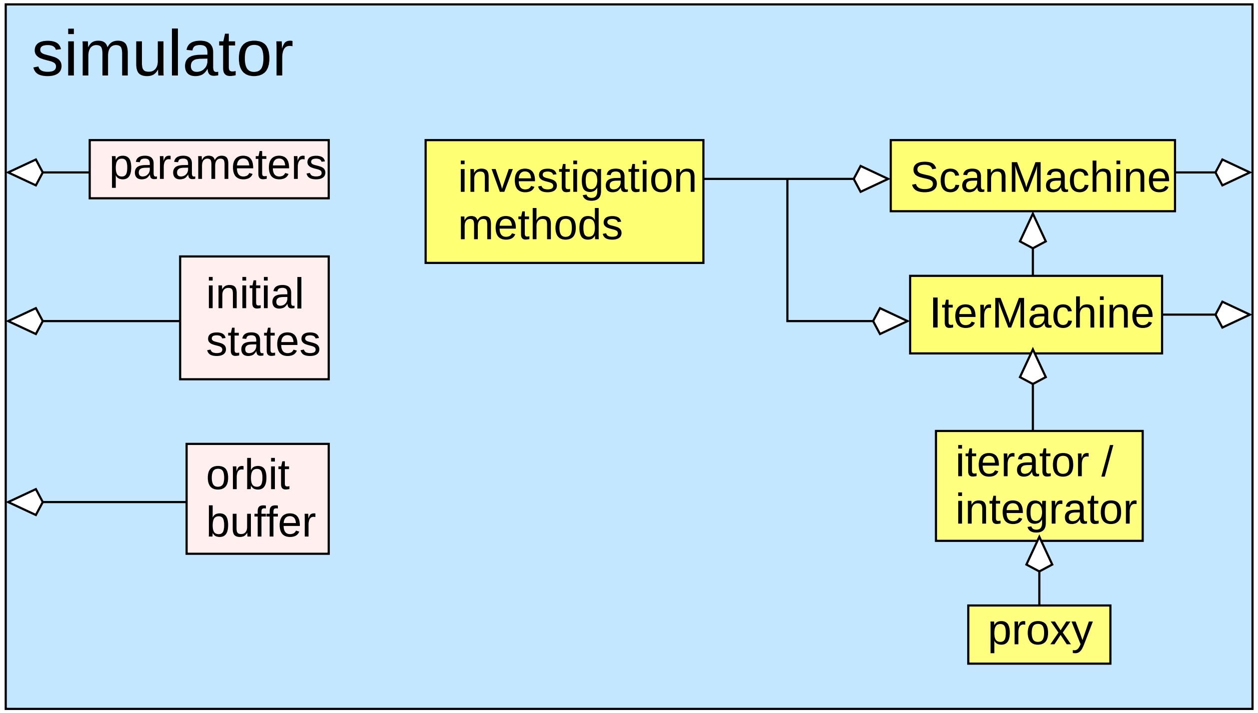
SCAN MACHINE



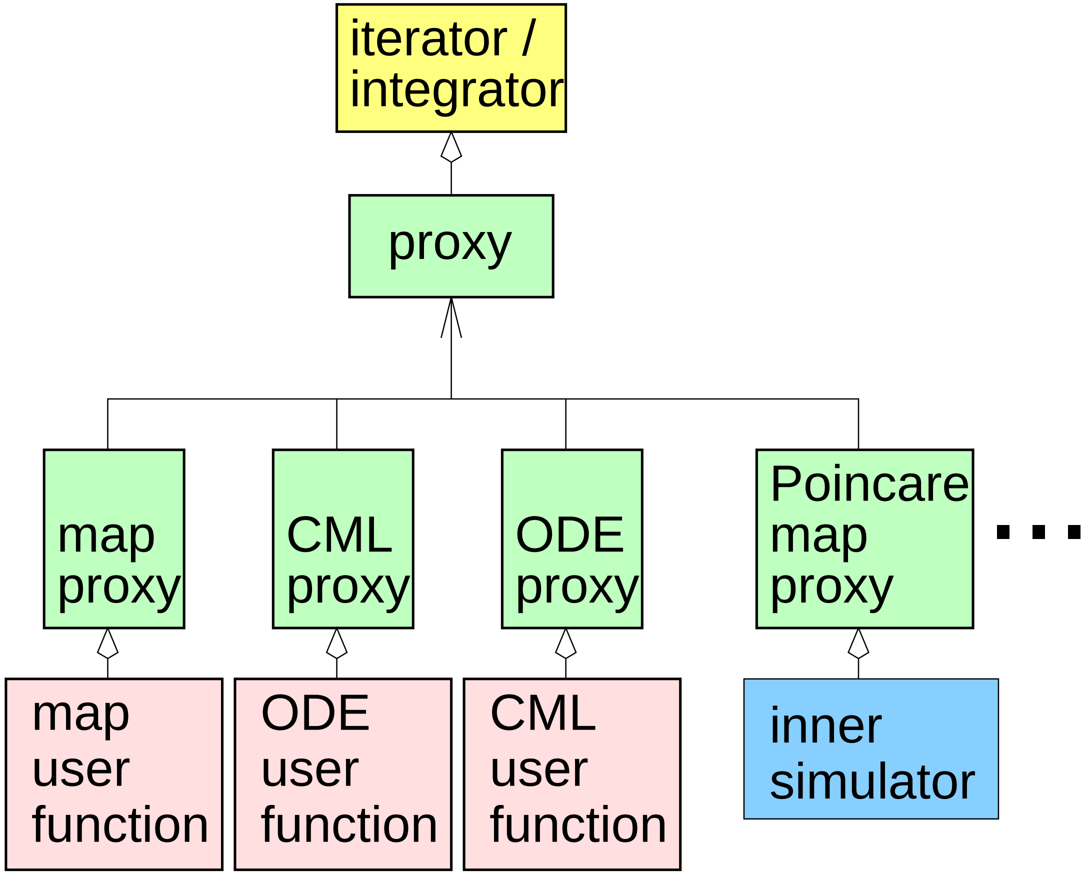
SCAN MACHINE



Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)



POINCARÉ MAPS



Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)

- Using of hybrid automata is more suitable for modeling and model validation.
- Using of hybrid equations is more suitable for investigation of dynamics occurring in hybrid systems and allows easy to embed the simulation in the more global simulations context.

- more to **AnT 4.667** : http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/nld/software/