AnT 4.669 – a tool for simulating and investigating dynamical systems

V. Avrutin, R. Lammert, M. Schanz, G. Wackenhut

Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart, Universitätstrasse 39, D-70569 Stuttgart, Germany
{Avrutin,Lammert,Schanz,Wackenhut}@informatik.uni-stuttgart.de

Abstract

A software package for the simulation and investigation of the dynamic behavior of dynamical systems called AnT will be presented in this paper. Due to its flexible architecture, AnT is able to simulate dynamical systems belonging to various classes, e.g. maps, ordinary and delay differential equations, etc., as well as many sub–classes derived from these. A main feature aimed at the development of AnT is the support of the investigation of the dynamics of the simulated systems with several provided investigation methods, like e.g. period analysis, Lyapunov exponents calculation, generalized Poincaré section analysis and much more. Another important feature of AnT are so–called scan runs, i.e. the ability to investigate a dynamical system by varying some relevant influence quantities, such as the control parameters, initial values, or even some parameters of the investigation methods.

Key words: simulation, dynamical systems, non–linear dynamics, numerical investigation methods, distributed computing

1 Introduction

Efficient simulation of dynamical systems is an important field in research, industrial applications and education as well. The project presented here is focused on the development and implementation of a software package AnT for the simulation and investigation of a broad spectrum of dynamical systems.

The AnT simulator comes along with a set of investigation methods, thus enabling the user to analyze various aspects of different classes of dynamical systems. Hereby, the user must only provide an equation of motion corresponding to one of the supported classes of dynamical systems and do the appropriate initialization of the simulator, e.g. by means of a configuration file or, more comfortably, by using a corresponding graphical user interface. The ability to turn several investigation

methods either on or off leads to an increase in flexibility and also efficiency, since one only has to pay for resources that are really needed.

The AnT simulation package is a powerful tool which provides the user with data resulting from the performed computations. The user of the simulator may further process and interpret the produced data in order to get valuable insights concerning the dynamics of the investigated system.

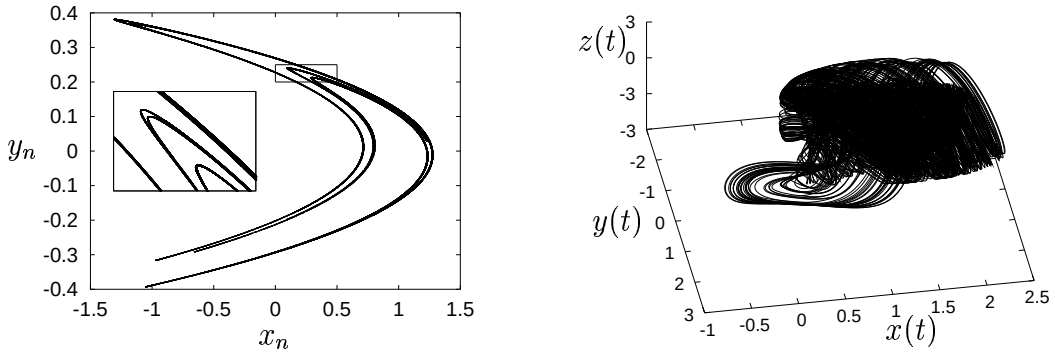
Compared to other known tools for the simulation and investigation of dynamical systems, like for instance AUTO, XPP, DsTool, Dynamics, DDE-BIFTOOL, AnT has some advantages. Firstly, the spectrum of dynamical system classes and investigation methods supported by AnT is broader than that of any other tool which we know so far. Secondly, AnT allows the investigation of dynamical systems while varying some settings. Although other tools are also able to perform similar tasks, AnT is by now the only tool running in client-server mode. Hence, computation intensive tasks may be distributed among many clients automatically.

2 Classes of dynamical systems supported by AnT

According to the aims of the AnT project, the term of a dynamical system was kept very general. In this section we present a brief overview of several types of dynamical systems which can be simulated and investigated using AnT. Currently, the simulator is able to deal with the following *basic classes* of dynamical systems:

- standard **discrete maps**: $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \{\sigma\})$
- **ordinary differential equations** (ODE): $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \{\sigma\})$
- **delay differential equations** (DDE): $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{x}(t-\tau), \{\sigma\})$
- **functional differential equations** (FDE): $\dot{\vec{x}}(t) = \vec{\mathcal{F}}[\vec{x}_t, \{\sigma\}]$ with $\vec{x}_t(\theta) = \vec{x}(t+\theta)$, $\theta \in [-\tau, 0]$
- **partial differential equations** (PDE) with one spatial direction: $\frac{\partial}{\partial t}\vec{x}(q,t) = \vec{f}\left(\vec{x}(q,t), \frac{\partial}{\partial q}\vec{x}(q,t), \ldots, \vec{\sigma}\right)$ with a scalar spatial component q defined on the domain $[q_{\min}, q_{\max}]$.

Here $\vec{x}_n$ and $\vec{x}(t)$ denote the real–valued state vector of the system, σ a set of parameters, and τ the time delay. The vector $\vec{x}_t(\theta)$ is an element of the extended state space given by the space of vector–valued functions on the interval $[-\tau, 0]$ and $\vec{\mathcal{F}}$ is a non–linear functional defined on this space. Dealing with partial differential equations, one can use several types of von Neumann boundary conditions (fluxless, cyclic, interpolated, constant), as well as Dirichlet boundary conditions. Figures 1.a, 1.b, 2.a show some examples of attractors of dynamical systems belonging to these basic classes.



(a) discrete map

(b) ordinary differential equation

Fig. 1. *Attractors of some dynamical systems investigated using* AnT.

(a) *Hénon map* $x_{n+1} = 1 - ax_n^2 + y_n$, $y_{n+1} = bx_n$ *([3]). A standard example of a two–dimensional system discrete in time. Shown is the chaotic attractor at* $a = 1.4$, $b = 0.3$. *A small rectangular area of the state space is shown enlarged in the zoom window.*

(b) *Lorenz 84 system* $\dot{x} = aF - ax - (y^2 + z^2)$, $\dot{y} = -y + G + (xy - bxz)$, $\dot{z} = -z + (bxy + xz)$ *([4]). Presented is the chaotic attractor at* $a = 0.25$, $b = 4.0$, $F = 8.0$, $G = 1.0$.

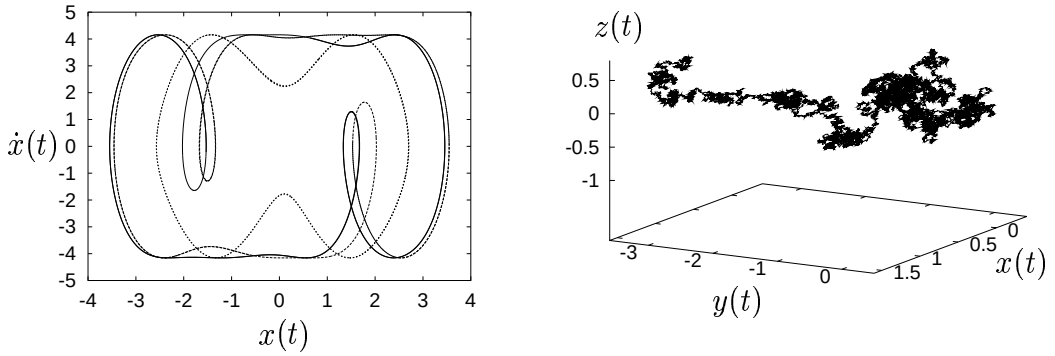
In addition to standard discrete maps it is also possible to use

- **recurrent maps** $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \vec{x}_{n-1}, ..., \vec{x}_{n-n_\tau}, \{\sigma\})$

with $n_\tau \in \mathbb{N}$, $n_\tau > 1$. Of course, the recurrent maps can be transformed into standard discrete maps by extending the state space accordingly. Hence, one is in principle always able to do the work with standard maps only, but using the more natural form of recurrent maps simplifies the modeling of time discrete systems with delay and is therefore supported by AnT. As a side note, we have to remark that this kind of elimination of the delay by state space extension is not possible for delay differential equations ([1], [2]).

Non–autonomous systems are not supported by AnT directly, but this is not a restriction, since they can be transformed into autonomous ones by a standard extension of the state space, i.e. by introducing a new state variable representing the time.

Additionally, AnT can be used with two more types of dynamical systems, denoted as *composite* or *cellular* dynamical systems. These are known as **coupled map lattices** (CML) and **coupled ordinary differential equations lattices** (CODEL) respectively. Given local couplings and closed ring topology, these systems can be defined by



(a) delay differential equation (b) stochastical ODE

Fig. 2. *Attractors of some dynamical systems investigated using* AnT.

(a) *Phase locked loop with normalized time delay* $\dot{x}(t) = R\sin(x(t-1))$ *([5]). The three attractors coexisting at* $R = 4.157$ *are shown here, a symmetric limit cycle and two asymmetric ones, symmetric to each other.*

(b) *Example of a stochastic system* $\dot{\vec{x}} = \mathbf{L}\vec{x} + \vec{\eta}$*. The figure illustrates a single realization of the trajectory of a particular three–dimensional Ornstein–Uhlenbeck process ([6]) using the matrix* $\mathbf{L} = a\mathbf{I}$ *with* $a = -10^{-3}$ *and* $\mathbf{I}$ *the identity matrix. Each component of the additive white noise vector* $\vec{\eta}$ *has Gaussian distribution* $N(0,1)$*.*

- CML: $\vec{x}_{n+1}^{(i)} = \vec{f}\left(\vec{x}_n^{(i-l) \bmod N}, \ldots, \vec{x}_n^{(i)}, \vec{x}_n^{(i+r) \bmod N}, \{\sigma\}\right)$
- CODEL:
  $\dot{\vec{x}}^{(i)}(t) = \vec{f}\left(\vec{x}^{(i-l) \bmod N}(t), \ldots, \vec{x}^{(i)}(t), \ldots, \vec{x}^{(i+r) \bmod N}(t), \{\sigma\}\right)$

Hereby, N denotes the number of cells in the lattice, i is the cell index ($i = 1 \ldots N$) and l and r the ranges of the coupling interval. AnT provides also more general types of lattices with global coupling, where a cell is coupled arbitrarily with other existing cells. Hence, arbitrary lattice topology is also possible, as well as lattices with more than one space direction. Examples of coupled systems are shown in Fig. 3 and Fig. 4.

Furthermore, AnT has support for *hybrid dynamical systems*, which became more and more important during the past few years. A basic property of these systems is, that their state space consists of two parts, a continuous–valued vector $\vec{x}_n \in \mathbb{R}^{n_c}$ and a vector $\vec{m} \in \mathbb{M}^{n_d}$ where $\mathbb{M}$ is a set of discrete values and $n_c, n_d \in \mathbb{N}^+$. AnT provides three types of hybrid systems, i.e.

- **hybrid maps**: $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \vec{m}_n, \{\sigma\}) \quad \vec{m}_{n+1} = \vec{\phi}(\vec{x}_n, \vec{m}_n, \{\sigma\})$
- **hybrid ODEs**:
  $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{m}(t), \{\sigma\}) \quad \vec{m}(t^+) = \vec{\phi}(\vec{x}(t), \vec{m}(t), \{\sigma\})$

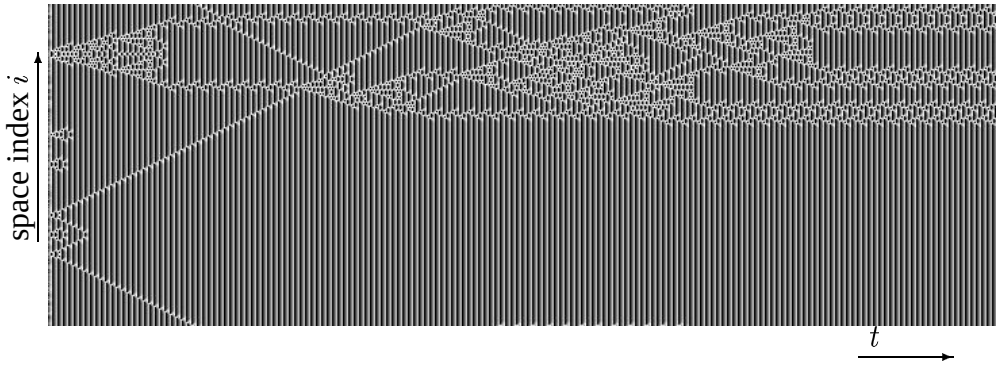


Fig. 3. *Dynamics of some cellular dynamical systems, supported by* AnT.

Coupled map lattices may show spatio–temporal structures, like the one presented here. The lattice is defined by

$$x_i(n+1) = f(\varkappa_i(n)) \quad \forall i = 1 \ldots N \quad f(x) = \begin{cases} x + a & \textit{if } x < 1 \\ 0 & \textit{if } x_n \geq 1 \end{cases}$$

$$\varkappa_i(n) = \begin{cases} \frac{\gamma_1 x_N(n) + \gamma_2 x_0(n) + \gamma_3 x_1(n)}{\gamma_1 + \gamma_2 + \gamma_3} & \textit{if } i = 0 \\ \frac{\gamma_1 x_{i-1}(n) + \gamma_2 x_i(n) + \gamma_3 x_{i+1}(n)}{\gamma_1 + \gamma_2 + \gamma_3} & \textit{if } 0 < i < N \\ \frac{\gamma_1 x_{N-1}(n) + \gamma_2 x_N(n) + \gamma_3 x_0(n)}{\gamma_1 + \gamma_2 + \gamma_3} & \textit{if } i = N \end{cases}$$

The gray scale values correspond to the values $x_i(n)$*. The lattice consists of* 238 *cells, randomly initialized, and locally coupled using the ring topology. Parameter settings:* $\gamma_1 = \gamma_2 = \gamma_3 = 1$*,* $a = 0.36$*. In the left part of the figure, before the stationary state is reached, the glider–phenomenon, known from the field of cellular automata, can be observed.*

- **hybrid DDEs**
  $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{x}(t-\tau), \vec{m}(t), \{\sigma\}) \quad \vec{m}(t^+) = \vec{\phi}(\vec{x}(t), \vec{m}(t), \{\sigma\})$

The notation $\vec{m}(t^+)$ means that the vector $\vec{m}$ is assumed to be left-side continuous with respect to time.

AnT is also able to deal with some types of *stochastic systems* with additive noise η, although it has to be remarked, that currently only a limited set of integration and investigation methods is available for these systems. The stochastic system types supported by AnT are:

- **maps with additive noise**: $\vec{x}_{n+1} = \vec{f}(\vec{x}_n, \{\sigma\}) + \vec{\eta}_n$
- **ODEs with additive noise**: $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \{\sigma\}) + \vec{\eta}(t)$
- **DDEs with additive noise**: $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{x}(t-\tau), \{\sigma\}) + \vec{\eta}(t)$

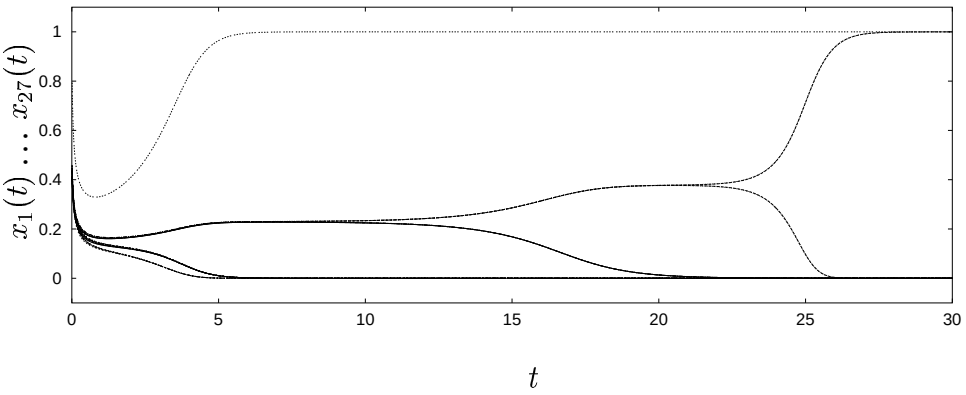


Fig. 4. *Dynamics of some cellular dynamical systems, supported by* AnT.

An example for a CODEL with global coupling is the following system of coupled selection equations:

$$\dot{x}_{ijk}(t) = x_{ijk}(t) \left(1 + (3\beta - 1) x_{ijk}(t)^2 - \beta \sum_{j'k'} x_{ij'k'}(t)^2 - \beta \sum_{i'k'} x_{i'jk'}(t)^2 - \beta \sum_{i'j'} x_{i'j'k}(t)^2 \right)$$

This system is developed based on pattern formation concepts, and is intended for solving a special problem of combinatorial optimization, the so–called three–index assignment problem ([7]). The time series presented here corresponds to the system with problem size 3*. The system has hence* $3^3 = 27$ *state variables. Parameter setting is* $\beta = 2$*.*

Here $\vec{\eta}$ means the additive white noise vector, which can be distributed differently, e.g., uniform or gaussian. An example of a stochastic ODE with additive noise is presented in Fig. 2.b.

A remarkable feature of AnT is the very general implemented concept of **Poincaré sections**. According to this concept, the orbit of a discrete map is resulting from the orbit of some other dynamical system, whereby only those points are selected which satisfy a specific condition. In the case of classical Poincaré sections, this condition is given by the cross–section of the orbit with a plane. AnT supports two variants of classical Poincaré sections, which behave differently in the context of scan runs. On the one hand, the plane can be fixed, i.e. the plane remains the same even if the parameters of the investigated dynamical system are varied. For instance, the behavior of the Rössler system ([8]) can be investigated by varying the parameter a using Poincaré sections with the fixed plane $y = 0$ (see Fig. 5). On the other hand, in some cases it can be interesting to define the plane depending on the system parameters, and hence the coefficients of the plane have to be varied together with the system parameters. For instance, it is known that the Lorenz–63

system ([9]) possesses three fixed points, which can be determined analytically. Two of these fixed points depend on the system parameters r and b. In the Fig. 6 one can see the bifurcation diagram of the Poincaré map, using the plane defined by the fixed points of the Lorenz–63 system. In addition to the classical Poincaré section approach, AnT can use any other condition in order to calculate generalized Poincaré sections. Especially, in the case of hybrid systems, there is a specific condition generating the Poincaré section, which is fulfilled whenever a change in the discrete state part occurs. Additionally, any user defined condition may be used as well. It has to be mentioned that, due to the architecture of AnT, Poincaré sections are represented as discrete maps with a special kind of system function, which iterates the given dynamical system until the Poincaré condition gets fulfilled. Hence, all investigation methods applicable for maps are also applicable for numerical calculated Poincaré maps.

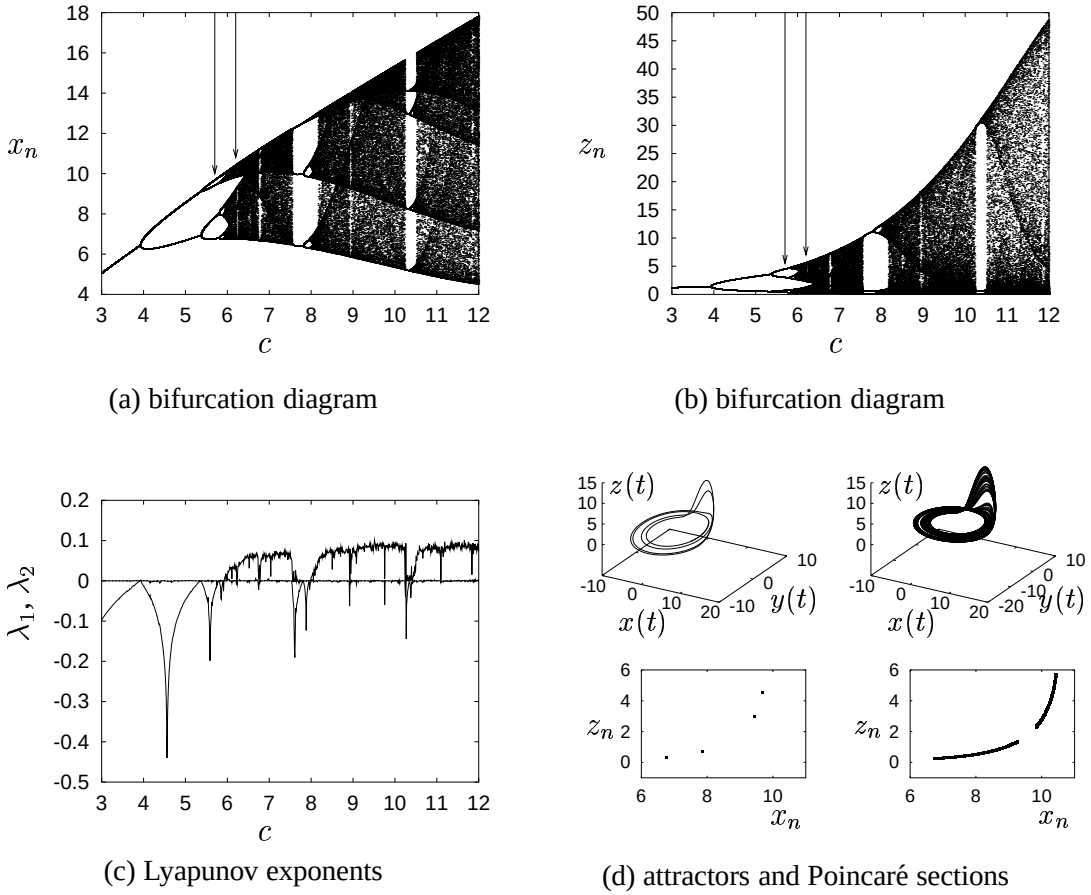
At last, we ought to remark that AnT can also proceed **external input data**. Although this topic was not intended to be the main application area of the AnT simulator, we were able to achieve this functionality by implementing our system function as an external data reader. Hence, this dynamical system belongs to the class of standard discrete maps and may be described informally by

- $\vec{x}_{n+1} =$ next input vector

Summarizing it has to be remarked that the support of such a broad spectrum of dynamical systems was not a primary aim in the early phases of the AnT project. However it turned out that the software architecture concepts, designed and used in the project (see sec. 6), allow this support generically. As a consequence hereof, we were able to implement many of our investigation methods at a high level of abstraction, thus reusing this functionality for many classes of dynamical systems and avoiding source code duplication.

3 Integration methods

In order to investigate dynamical systems continuous in time, i.e. ODEs, DDEs, etc., AnT integrates them numerically. Hereby, the continuous trajectory has to be approximated by a sequence of discrete states resulting from the applied integration method. To cope with typical problems arising by the numerical integration, like for instance by the integration of the so–called stiff differential equations or the integration of stochastical systems, AnT provides several classes of integration methods. Currently implemented methods with fixed step size include the well–known one–step methods of Runge–Kutta type defined by their corresponding pre–implemented Butcher arrays. Additionally the Butcher arrays can be also supplied by the user. Hence, AnT supports user defined integration methods of this type. Furthermore a large collection of different integration methods is implemented, con-



(a) bifurcation diagram

(b) bifurcation diagram

(c) Lyapunov exponents

(d) attractors and Poincaré sections

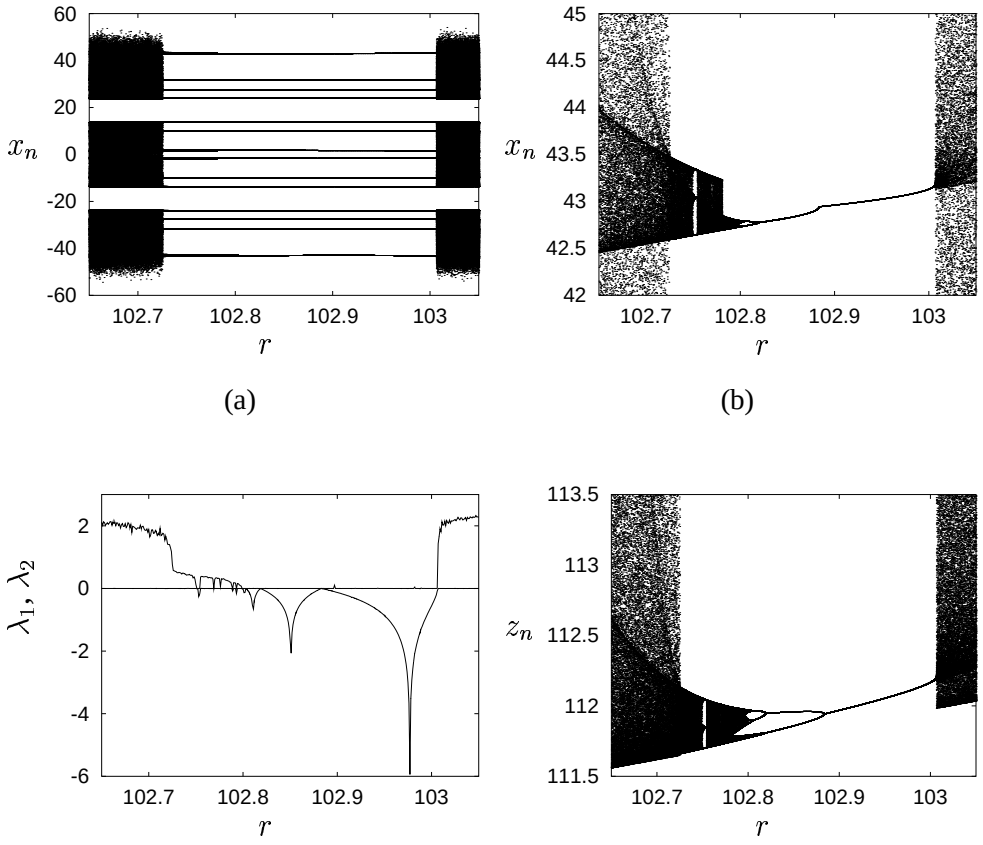
Fig. 5. *Investigation of the Rössler system.*

The Rössler system, given by $\dot{x}(t) = -(x(t) + z(t))$, $\dot{y}(t) = x(t) + ay(t)$, $\dot{z}(t) = b + z(t)(x(t) - c)$, *is here investigated using the parameter settings* $a = 0.15$, $b = 0.2$ *and varying* c.

(a),(b) *Bifurcation diagrams for the Poincaré map, defined by the cross–sections of the trajectory with the half–plane* $\left\{(x, y, z)^T \mid y = 0,\ x > 0\right\}$.

(c) *The figure shows the largest two Lyapunov exponents of this three–dimensional system continuous in time.*

(d) *A periodic and a chaotic (two–band) attractor and the corresponding Poincaré sections. The parameter values used here (*$c = 5.7$ *and* $c = 6.2$*) are marked with arrows in the bifurcation diagrams shown in (a) and (b).*



(a) (b)

(c) Lyapunov exponents (d)

Fig. 6. *Investigation of the Lorenz–63 system.*

The Lorenz–63 system, given by $\dot{x}(t) = \sigma(y(t) - x(t))$, $\dot{y}(t) = x(t)(r - z(t)) - y(t)$, $\dot{z}(t) = x(t)y(t) - bz(t)$ *is here investigated using the parameter settings* $\sigma = 16.0$, $b = 4.0$ *and varying* r. *Here, a small periodical window within the chaotic regime can be observed.*

(a) *Bifurcation diagram for the* x *variable of the Poincaré map, which is defined by the cross–section of the trajectory with the parameter–dependent plane defined by the three fixed points of the system* $O = (0, 0, 0)^T$ *and* $P^{\pm} = \left(\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1\right)^T$.

(b), (d) *Parts of the bifurcation diagrams for the* x *and* z *variables of the Poincaré map mentioned above.*

(c) *Largest two Lyapunov exponents.*

taining among others some implicit methods like Heun–backward, some multi–step methods with memory like the backward differentiation approach (Gear method) and some predictor–corrector methods like the Adams–Moulton approach. Additionally, AnT supports several methods with adjustable step size.

4 Scans

An important issue within the theory of non-linear dynamical systems is the investigation of the system dynamics depending on various settings. AnT can handle this kind of investigation by allowing so–called scan–runs, which incorporate the analysis of the system dynamics for varying settings. During such a scan–run, not only one or more parameters of the investigated dynamical system can be varied, but also the initial values as well as specific parameters of the applied investigation methods. One possible application of the scan–run is the investigation of bifurcation scenarios showing the dependency of the system dynamics on some control parameters. AnT supports one–dimensional scan–runs, where only one control parameter is varied (see Fig. 5, Fig. 6 and Fig. 8), as well as higher–dimensional scans (see Fig. 9). Another domain of application for scan–runs is the investigation of basins of attraction in the case of coexisting attractors. Furthermore, one can empirically determine the parameter settings of certain investigation methods for which these methods operate optimally. A sample application of this case is presented below, but we have to remark that such scans are rather unusual, since scan–runs normally concern the parameters or initial values of the system.

One of the approaches for the calculation of the Lyapunov exponents, implemented by AnT (some modification of the approach of Wolf *et al.* [10]), possesses as parameters the length of a small deviation vector $|\vec{\varepsilon}|$ and the number N_{GSO} of iteration steps between two subsequent applications of the Gram–Schmidt ortho–normalization procedure. These parameters must be provided by the user. In order to determine the suitable settings for the parameters, one can investigate their influence on the calculated values of Lyapunov exponents by using them as scan parameters in a scan–run. In the example presented in Fig. 7, the correct value of the Lyapunov exponent can be determined analytically, which yields $\lambda^* = \ln(2)$. In Fig 7.a, the accuracy of the numerical calculation is shown depending on the length of the deviation vectors $|\vec{\varepsilon}|$. The Gram–Schmidt ortho–normalization procedure is applied in each step. As one can see, the appropriate setting for $|\vec{\varepsilon}|$ lies between $\varepsilon_{\min} \propto 10^{-9}$ and $\varepsilon_{\max} \propto 10^{-7}$. If one sets $|\vec{\varepsilon}| \gg \varepsilon_{\max}$, the results are not precise enough due to approach specific requirements, by $|\vec{\varepsilon}| \ll \varepsilon_{\min}$ the accumulated numerical errors made by the computer become perceptible. By increasing the number of iterations, one achieves more precise results, but the suitable ranges for ε will be smaller. In Fig. 7.b, the accuracy of the numerical calculation is shown depending on the number of iteration steps between two subsequent applications of the Gram–Schmidt ortho–normalization procedure. As one can see, the best re-

sults are achieved, if the procedure is applied in each step. We remark that for other dynamical systems, especially continuous in time, the optimal setting for the parameter N_{GSO} may be larger, depending on the step size used by the numerical integration method.

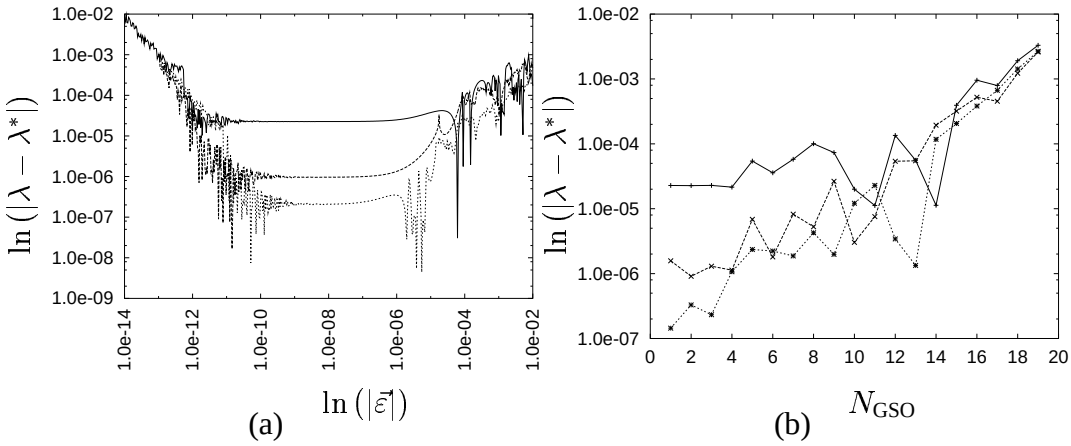


Fig. 7. *Accuracy of the numerical calculation of the Lyapunov exponents for the logistic map* $x_{n+1} = \alpha x_n(1 - x_n)$ *for* $\alpha = 4.0$*. The number of iterations is* $25\,000$ *steps (solid line),* $250\,000$ *steps (dashed line) and* $2\,500\,000$ *steps (dotted line). The transient part is* 1000 *steps in all three cases. The initial value is* $x_0 = 0.1$*.*

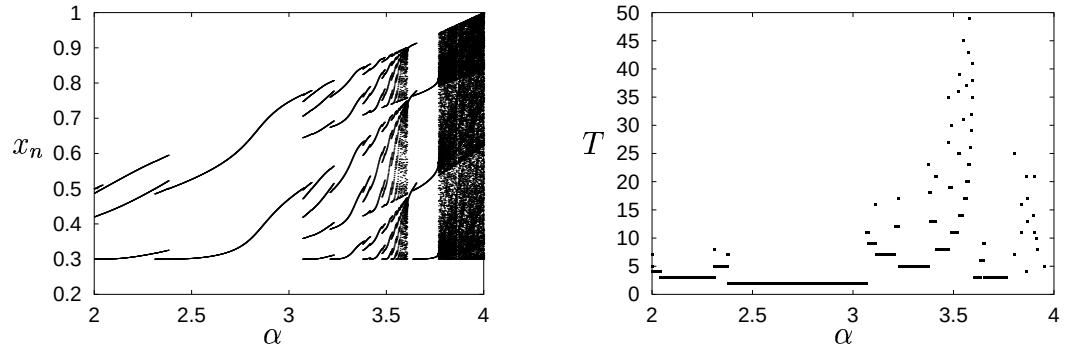
5 Investigation methods

As already mentioned, AnT provides a lot of investigation methods, considering various aspects of the behavior of a dynamical system. In order to acquire a rough idea of the investigated dynamical system, one can use the basic trajectory evaluations. For instance, one can calculate one or more trajectories of the system, the orbital and average velocities as well as other values like e.g. the wave numbers ([**?**], Fig. 8.d). The period analysis makes it possible to calculate bifurcation and period diagrams. This is particularly useful in combination with one– or higher–dimensional scan runs (compare Fig. 8.a, Fig. 8.b and Fig. 9). The stability properties of a dynamical system can be investigated by calculating the Lyapunov exponents of its attractors (Fig. 7, Fig. 8.c). This investigation method is implemented not only for maps and ordinary differential equations, but also for dynamical systems with memory, such as delay differential equations. Due to its generality, this method can also be applied to the more general class of functional differential equations.

We have included some standard techniques from the field of frequency analysis, like e.g. the calculation of the power spectra. Singular value decomposition is another useful method provided by AnT.

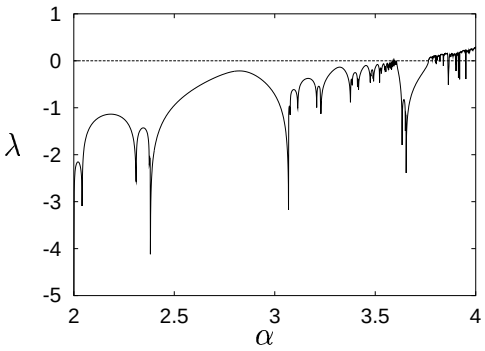
Methods based on symbolic dynamics provide a collection of generic techniques for the generation of the symbolic image of an underlying trajectory. From the symbolic image, one can obtain some quantitative measures like e.g. its entropy, which can be used to estimate some properties of the underlying trajectory.

The results of each investigation method are written to corresponding data files. After a simulation run, these results can be visualized and evaluated by the user. Additionally, 2D and 3D online-visualization modules are implemented, which can be used during single-run simulations.

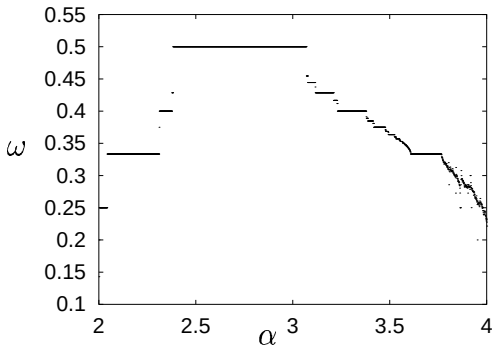


(a) bifurcation diagram

(b) period diagram



(c) Lyapunov exponent λ



(d) wave number ω

Fig. 8. *Some investigation results.*

Presented are some aspects of the dynamical behavior of the following one–dimensional dynamical system discrete in time, with piece–wise defined system function

$$x_{n+1} = \begin{cases} \alpha x_n(1-x_n) & \text{if } x_n \leq \frac{1}{2} \\ \beta x_n(x_n-1)+1 & \text{if } x_n > \frac{1}{2} \end{cases} \tag{1}$$

at $\beta = 2.8$ *and varying values of* α*.*

(a) *Bifurcation diagram. Shown are the periodic and aperiodic dynamics of the system.*

(b) *From the period diagram one can see that the bifurcation scenario presented in Fig.* (a) *resembles a period adding scenario [12].*

(c) *As one can see from the diagram of the Lyapunov exponent, the bifurcations presented in Fig.* (a) *take mostly place on super–stable parameter values.*

(d) *In the wave number diagram one can recognize the typical "devil's staircase" structure [13].*

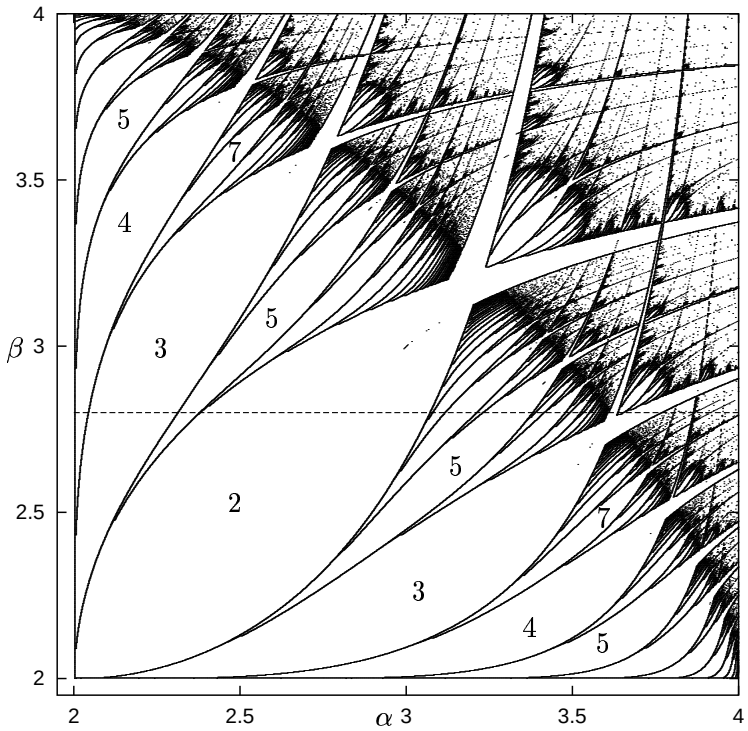


Fig. 9. *Bifurcation structures of the system (1) in the 2-dimensional parameter space* $[\alpha \times \beta]$.

The white spaces denote regions with the same periodical behavior, calculated by the method denoted as region analysis. Some of the periods are written within the corresponding regions. The dashed line shows the section corresponding to a 1-dimensional parameter scan for α *with* $\beta = 2.8$*, as presented in Fig. 8.*

6 Architecture

AnT is a very powerful tool due to its underlying software concepts. An important issue concerns e.g. the selective insertion of investigation methods during the initialization phase. Hereby is to mention, that an investigation method is implemented as consisting of executable entities spread all over the program but cooperating while performing the joint task.

In order to make this executable entities pluggable and mutually exchangeable we designed a common abstract base class denoting an abstract state transition. Using the main concepts of polymorphism and inheritance well–known from the field of the object–oriented programming we derived several concrete state transitions. Typical examples hereof are the calculation of the next state vector as performed by any iterator or integrator, the execution steps of some investigation method, and so on. For reasons of flow control, we were interested in finding a possibility to join these executable entities together, thus building so–called state machines. We developed some common types of state machines for general use, including e.g. transition sequences as well as cyclic state machines, which perform an inner transition as long as a given condition holds. Another kind of state machine heavily used is that of a pre–post state machine, consisting of three transitions executed sequentially and corresponding to the initialization, main computation and finalization phase of an algorithm.

The iterative nature of the simulation of dynamical systems implied the use of a special kind of state machine in order to perform the iteration steps. This iteration machine is a pre–post state machine, whereby the main computation is performed by a cyclic state machine. Its main task is to calculate the trajectory of the simulated system, however there are also other activities being performed as well, such as e.g. transitions representing parts of investigation methods (so–called plug–ins). This state machine terminates usually as soon as a certain number of iteration steps is reached.

The iteration machine described above is contained in the so–called scan machine, which has a similar structure, namely a pre–post state machine containing a cyclic state machine. As in the case of the iteration machine the main job was to calculate the trajectory of the system, so is the most important part of the scan machine even the execution of the embedded iteration machine. This will be repeated many times, since the iteration machine will be executed within the cyclic part of the scan machine as long as the scan condition holds.

According to the architecture presented above, an investigation method is implemented as a collection of state transitions. These have to be inserted into the right place, i.e. either into the iteration or into the scan machine, depending on the task to be performed within the specific transition. For instance, methods producing some

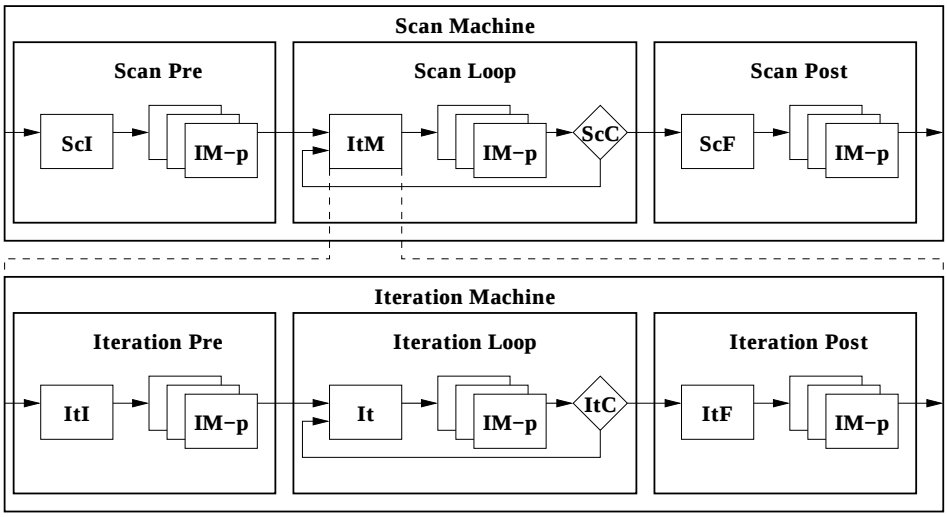


Fig. 10. *Architecture of the* AnT *simulator.*

ScI	*initialization of the scan machine*	ItI	*initialization of the iteration machine*
ScC	*scan condition check*	It	*iterator*
ScF	*final part of a scan run*	ItC	*iteration condition check*
ItM	*iteration machine*	ItF	*final part of an iteration run*
IM-p	*investigation method plug–in*		

data files must have at least two transitions related to opening and closing the corresponding files, mostly along with another state transition which performs the real computation and output. Usually, the opening act belongs to the pre–part of the involved state machine (e.g. the scan machine) while the closing of a file should be done within the post-transition.

7 Distributed run mode

An important feature of AnT is the possibility of automatically distributing scan-runs among several nodes (processors, machines), thus executing the simulations simultaneously (currently this feature is available only for Linux/UNIX platforms).

The parallel execution of a scan-run is coordinated by a server, which provides the running clients with the values of the scan settings to proceed with. In order for a client to do the required computation, it must be initialized by the first interaction with the server. Thereafter, it enters a cyclic process of requesting scan points from the server, doing the calculations it was initialized for, and finally, transferring the results back to the server, which has to perform the whole data management (e.g.

writing the incoming data out to files in a consistent manner). The advantage of this configuration is that if a client fails, the server will not be affected and the computation will be performed by the other clients. Another advantage is that clients can join the computation dynamically, so if computing power is getting available new clients can be added to the already running computation.

It is worth to mention that a client has to do quite the same work as in the case of a stand–alone application, with only minor changes concerning especially the output, which has to be redirected to the appropriate network connection (socket). From a user's viewpoint, the server is just another application implemented in order to control the interaction with its clients, as sketched above. Since the server is performing a scan run, we were able to realize it as a special kind of scan machine having no iteration machine inside.

8 Conclusion

The AnT simulator is free software and is distributed under the GNU public license. It will be available at the URL `http://www.AnT4669.de`. There will also be a more detailed description of the package.

The simulator is yet available for UNIX-based operating systems like Linux, Solaris or FreeBSD. Additionally, a port to the Windows platform is in work. The simulator behaves quite well on standard PC's (even on a 200 MHz Pentium-I processor) and workstations as well.

The simulation package uses free libraries only. For instance, in some numerical investigation methods the libraries fftwlib (`http://www.fftw.org`) and CLAPACK (`http://www.netlib.org/clapack`) are used. If these libraries are not installed on the current system, then the corresponding numerical methods (Fourier analysis, principal values decomposition) can not be loaded. However, the simulator works even without them, since the other methods of the simulator are not affected anyway.

For future releases, it is planned to extend the AnT simulator with new system types. Especially, partial differential equations (PDE) with more then one spatial direction and coupled delay and functional differential equations (CDDEL and CFDEL) will be considered. Another important goal is the extension of the list of supported investigation methods. Currently, we are working on more precise methods for the numerical calculation of entropies and several fractal dimensions and on including some new ideas from the field of symbolic dynamics, etc. Due to the simplicity of integration of new software modules into the AnT simulator, external researchers are welcome to contribute to the further development of AnT.

9 Acknowledgments

The authors wish to thank to the people who were involved in the development process of AnT. Many thanks to Prof. Levi (University of Stuttgart, Germany) for supporting us by our work.

References

[1] N. Krasovskii. *Stability of Motion*. Moskau, 1959, Translation: Stanford University Press, 1963.

[2] J.K. Hale. *Theory of Functional Differential Equations*. Springer-Verlag, 1977.

[3] M. Hénon. A two-dimensional mapping with a strange attractor. *Commun. Math. Phys.*, 50:69, 1976.

[4] E.N. Lorenz. Irregularity: A fundamental property of the atmosphere. *Tellus A*, page 36, 1984.

[5] W. Wischert, A. Wunderlin, A. Pelster, M. Olivier, and J. Groslambert. Delay-induced instabilities in nonlinear feedback systems. *Physical Review E*, 49(1):203–219, January 1994.

[6] F. Moos and P.V.E. McClintock, editors. *Noise in nonlinear dynamical systems*, volume 2: Theory of noise induced processes in special applications. Cambridge University Press, 1989.

[7] J. Starke and M. Schanz. Dynamical system approaches to combinatorial optimization. In D.-Z. Du and P. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 2, pages 471 – 524. Kluwer Academic Publisher, 1998.

[8] O.E. Roessler. An equation for continuous chaos. *Phys. Lett. A*, 57:397–398, 1976.

[9] E.N. Lorenz. Deterministic non-periodic flows. *J. Atmos. Sci.*, 20:130, 1963.

[10] A. Wolf, J.B. Swift, H.L. Swinney, and J.A. Vastano. Determining Lyapunov exponents from a time series. *Physica D*, 16:285–317, 1985.

[11] E.J. Ding. Wave number and symbolic dynamics. *Phys. Rev. A*, 39(2):816–829, 1989.

[12] V. Avrutin, G. Wackenhut, and M. Schanz. On dynamical systems with piecewise defined system functions. In *Proc. of Int. Conf. "Tools for Mathematical Modelling" MATHTOOLS'99*, 1999.

[13] P. Cvitanović, M. Jensen, L. Kadanoff, and I. Procaccia. Renormalization, Unstable Manifolds, and the Fractal Structure of Mode Locking. *Phys. Rev. Letters*, 55(4):343–346, 1985.