M.Biehl

Institute for Mathematics and Computing Science,

I. Nonlinear maps and chaotic dynamics

The (first) death of Laplace's Demon

LAPLACE (1749-1829):
assume an *intelligent, computing creature* exists, that – at a given time –

- has complete knowledge of the state of the world, i.e. knows the position and velocity of every particle in the universe
- knows the laws of nature which govern the motion of these particles
- can thus compute and predict the future of the world in every little detail

no room for: randomness, the free will, responsibility, guilt, merit ...
God(s) : spectator(s) who might have fixed initial conditions

apart from some *practical* problems (storage, computation),
Laplace's demon has died several times ... (relativity, quantum mechanics)

Poincaré (1854-1912):

(even very simple) mechanical systems

- can exhibit very *complex* behavior
- may be extremely sensitive to small changes in the initial conditions
- might react in a highly non-linear manner to such changes
  (small causes $\rightarrow$ large consequences)
- therefore, cannot be predicted over *long* time intervals
  (not even with more and more accurate measurements of the present state)

such systems are nowadays called **chaotic**

Even in a classical, Newtonian world **Laplace's demon fails**

He/she cannot even predict a simple driven pendulum (sec. I.3)

(not to mention the weather forecast...)

I. Nonlinear maps and chaotic dynamics

I.1. The logistic map

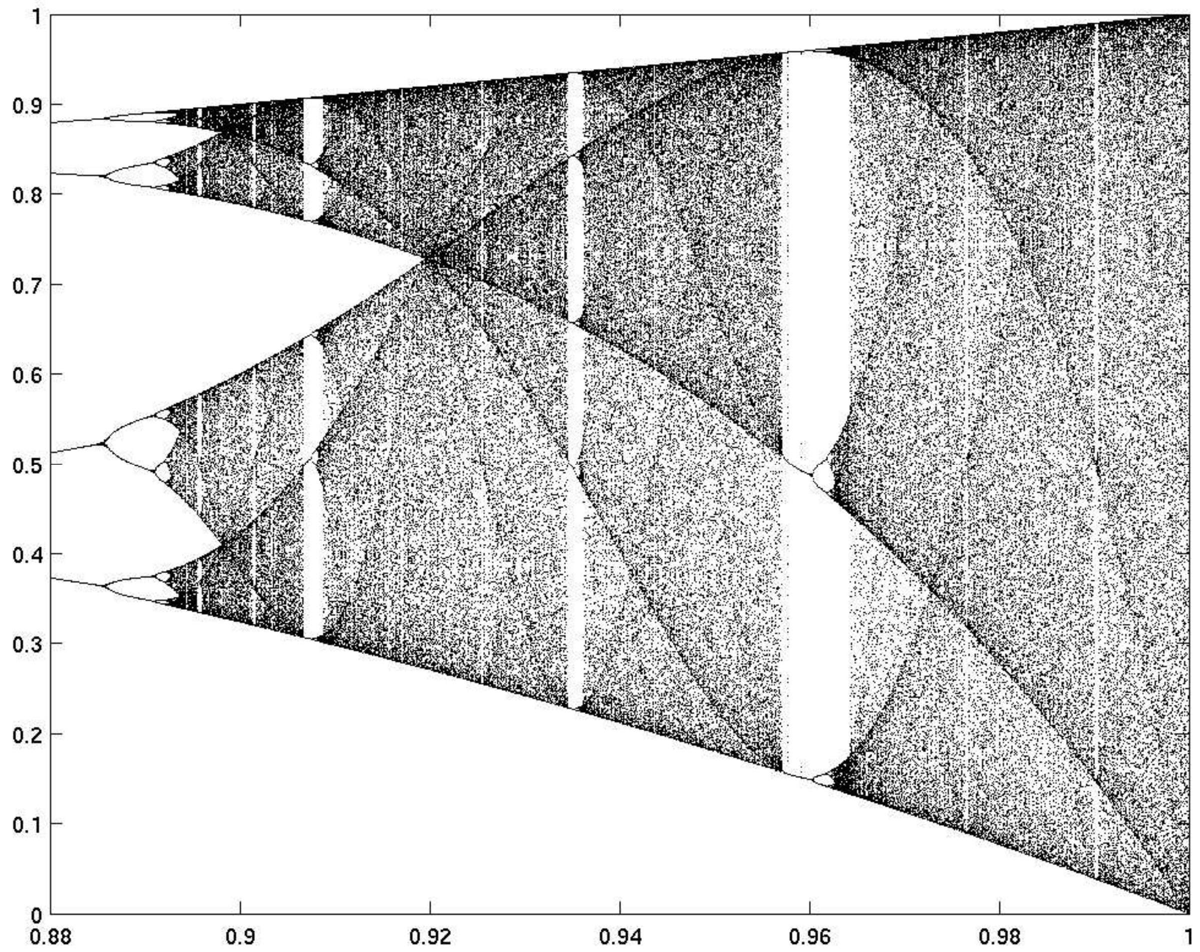
non-linear, one-dimensional discrete iteration which illustrates several features of chaotic systems (period doubling, Lyapunov exponent)

I.2. Random number generators

application of non-linear maps for generating pseudo random numbers (basis of all Monte Carlo methods, see secs. 2,3)

I.3. The chaotic pendulum a simple pendulum, subject to an external periodic torque, displays highly complex, chaotic behavior (ordinary differential eqs., *Poincaré section*)

1. The logistic map



(portion of the bifurcation diagram, as explained below)

The discussion follows closely the chapter *Population Dynamics* in W. Kinzel and G. Reents, *Physics by Computer* (Springer, 1998).

The logistic map

iteration of $x \in [0,1]$ with parameter $r \in [0,1]$: $\boldsymbol{x_{n+1} = f(x_n) = 4\, r\, x_n\, (1 - x_n)}$

- P.F. Verhulst (1845)
  a (very) simple model of a single species population dynamics

$x \equiv$ population density, change from year to year: $x_{n+1} - x_n = (4r-1)\, x_n - 4r\, x_n^2$

linear growth term (reproduction $\propto x$): $(4r-1)\, x = \begin{cases} < 0 \text{ for } r < 1/4 & \text{extinction} \\ > 0 \text{ for } r > 1/4 & \text{exp. growth} \end{cases}$

non-linear restriction (e.g. limited food supply): $-4\, r\, x^2 < 0$
hinders unlimited growth, prevents “population explosion”

- Prototype of a chaotic map, similar to those obtained from *Poincaré sections* of mechanical systems. Example: driven pendulum with friction, see sec. I.3.

Fixed points and periodic attractors

one obvious fixed point: $f(0) = 0$, $x = 0$ is reproduced in the iteration.

further observation: $$\lim_{n\to\infty} x_n = \begin{cases} x^* = 0 & \text{for } r < r_o = 1/4 \\ x^* > 0 & \text{for } r_o < r < r_1 \end{cases}$$

when is a fixed point stable/unstable (attractive/repulsive)?

formally: consider small deviation from a candidate fixed point $x^* = f(x^*)$

$$x_0 = x^* + \epsilon_o, \quad x_1 = f(x^* + \epsilon_o) \approx f(x^*) + \epsilon_o f'(x^*) = x^* + \underbrace{\epsilon_o f'(x^*)}_{\epsilon_1}, \quad f'(x) = 4r(1-2x)$$

after n iterations: $$\epsilon_n \approx f'(x^*)\,\epsilon_{n-1} \approx \left[f'(x^*)\right]^n \epsilon_o = \begin{cases} \to 0 & \text{for } |f'(x^*)| < 1 \\ \text{grows} & \text{for } |f'(x^*)| > 1 \end{cases}$$

Matlab routines

```
% logistic map
% the iteration  x(i+1) = 4 r x(i) (1- x(i))
% is implemented and can be studied by the
% following scripts and functions:
%
```

```
  function s = logstep(r,x,n)

% logstep.m defines the logistic map
% logstep(r,x,n) iterates 4*r*x*(1-x) n times
% beginning with initial value x

  for i=1:n
       x = 4 * r * x * (1-x);
  end
  s=x;
```

```
   function fp = fplot(r,n)

% fplot(r,n) plots the n-fold iteration  of the
% logistic map with paramter r (default: n=1)
% as x(i+1) vs. x(i)
% (includes  f(x)=x for n=0)

   if nargin==1
     n=1;
   end
```

```
   x = 0:0.001:1;
   y = x;
   for i =1:n
        y = logstep(r,y,1) ;
   end

   plot(x,y,'k-'); hold on;
   plot(x,x,'k-'); hold off;
```

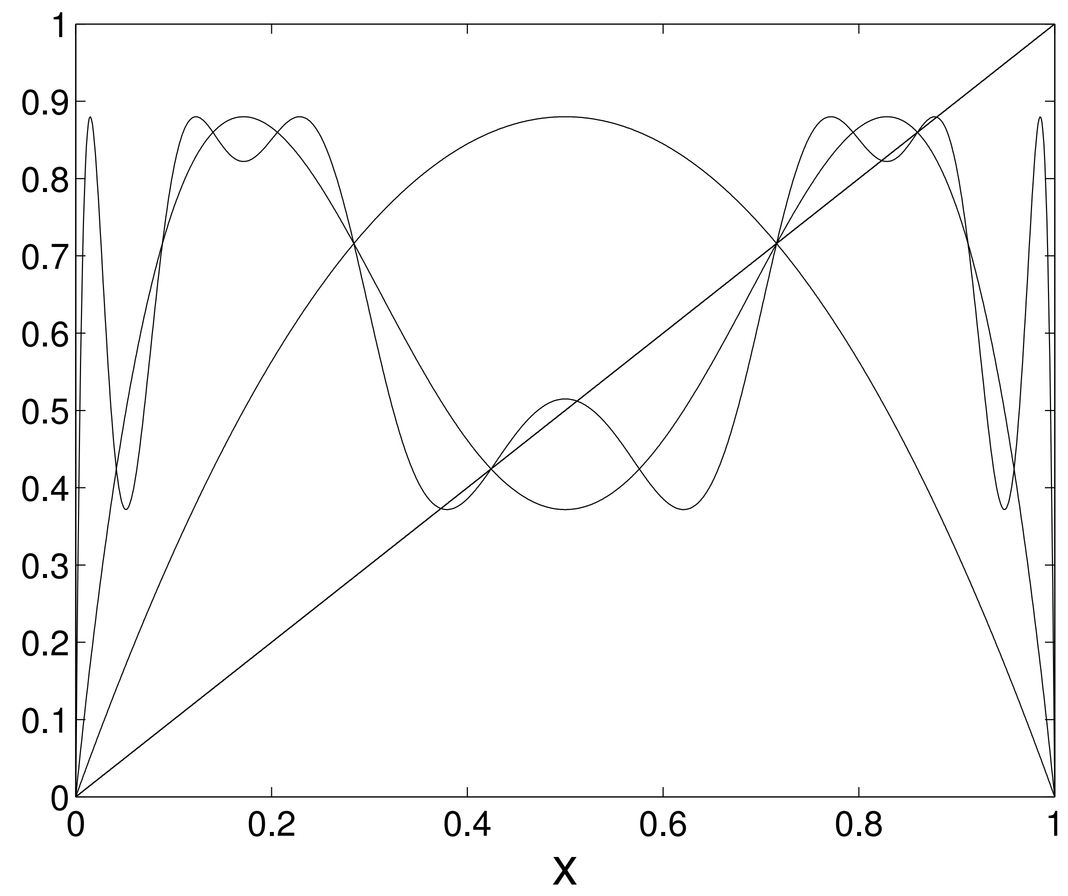


Fig. 2: *The logistic map $f(x) = 4rx(1-x)$, the two-fold iterated $f^{(2)}(x) = f(f(x))$ and $f^{(4)}(x)$ for $r = 0.88$. Intersections with $y = x$ correspond to stable or unstable fixed points.*

```
function s = iterplot(r,xinit,nsteps)

% function iterplot(r,xinit,nsteps)
% performs nsteps iterations of the logistic map
% with parameter r and plots
% a)  the x-values vs. "time"
% b)  the sequence of x-values as x(i) vs x(i-1)

  x= xinit * ones(1,nsteps);
  xprev = x;
  time = 1 : nsteps;

  for i=2:nsteps
     x(i) = logstep(r,x(i-1),1);
     xprev(i) = x(i-1);
  end

  figure(1);
  plot(time,x,'k-*')
  title([' r = ' num2str(r,'%0.4f')],'FontSize',18);
  axis([ 0 nsteps+1  -0.01  1.01 ]) ;
  axis square;

  figure(2);
  x1= ones(1,2*nsteps-2);  x2= ones(1,2*nsteps-2);

  for i=1:nsteps-1
       x1(2*i-1) = x(i); x2(2*i-1) = x(i);
       x1(2*i)   = x(i); x2(2*i)   = x(i+1);
  end
  plot(x1(2:2*nsteps-2),x2(2:2*nsteps-2));
  hold on;
  fplot(r,1);
  title([' r = ' num2str(r,'%0.4f')], 'FontSize',18);
  axis square; hold off;
```

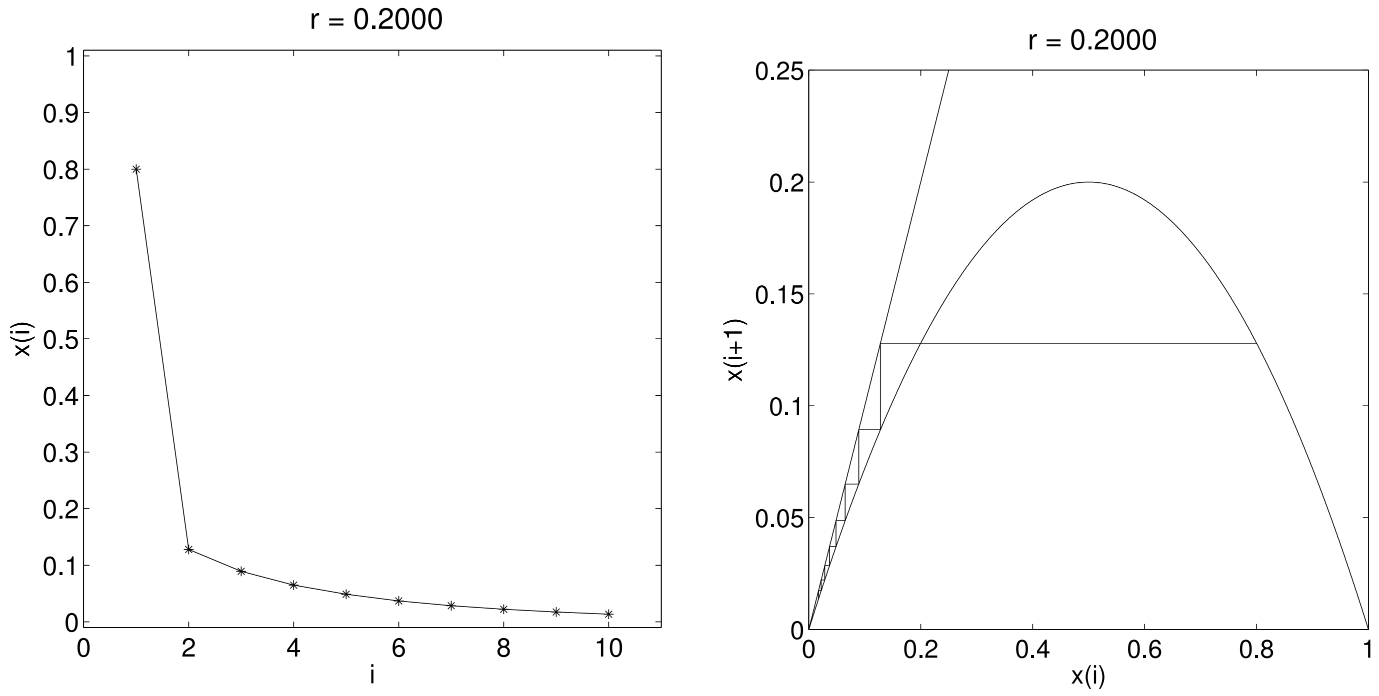


Fig. 3: *The iteration for $r = 0.2$ converges to $x = 0$*

Convergence to a single, non-zero fixed point

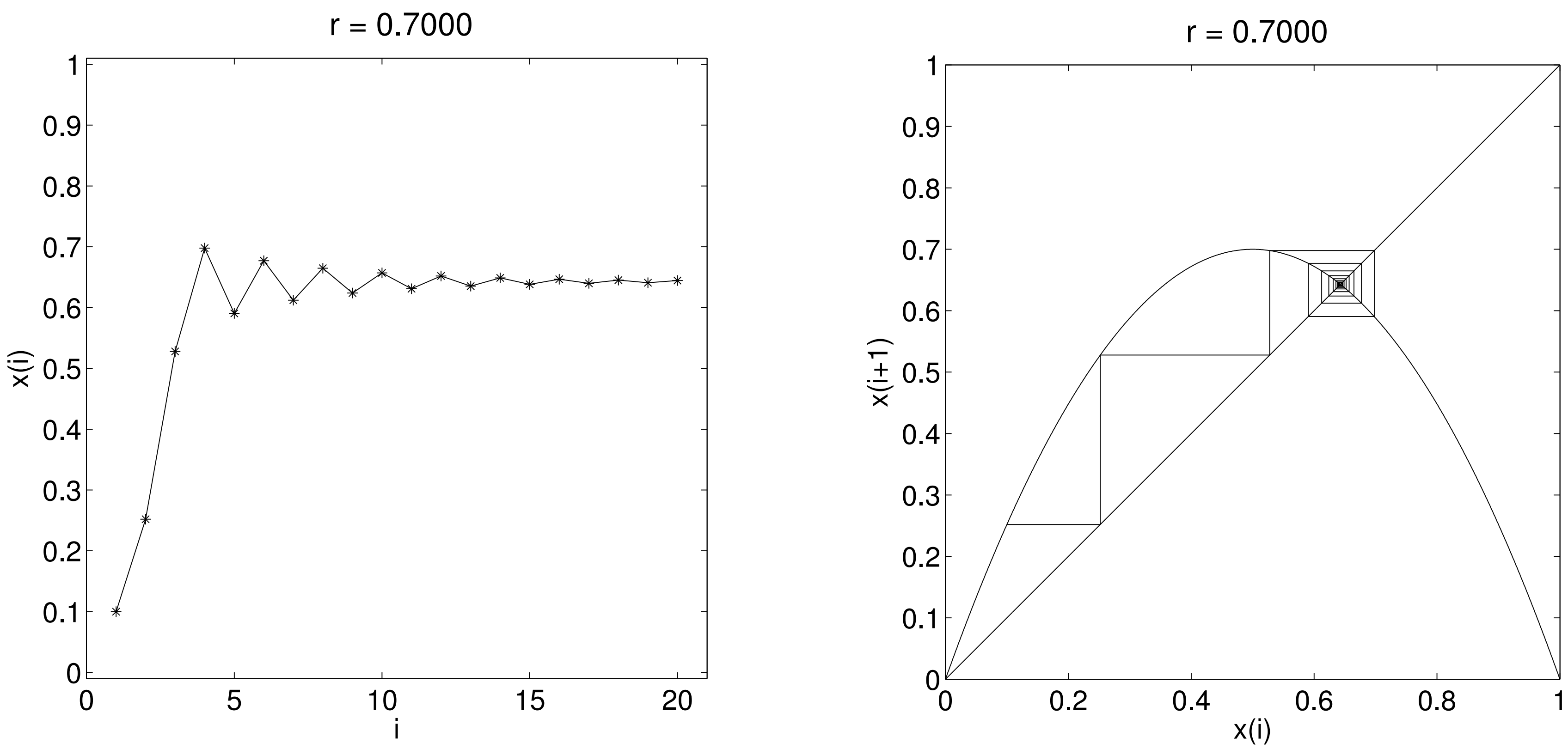


Fig. 4: *Iteration with $r = 0.7$: the f.p. $x = 0$ is unstable and x_n tends to $x^* \approx 0.64$ even from small initial values. Note that $|f'(0)| > 1$ whereas $|f'(x^*)| < 1$; as $f'(x^*) < 0$, the sign of the deviation from x^* alternates.*

single non-zero f.p.: $x^* = 1 - \frac{1}{4\,r}$

corresponding derivative: $|f'(x^*)| = |2 - 4\,r| = \begin{cases} < 1 & \text{for } r < r_1 = 3/4 \\ > 1 & \text{for } r > r_1 \end{cases}$

for $r > r_1 = 3/4$ all single f.p. are unstable!

The route to chaos: period doubling

for $r \gtrsim 3/4$ the simple fixed point is replaced

– by an attractor with period two: $x_1^*, x_2^*,$ with $f(x_1^*) = x_2^*$ and $f(x_2^*) = x_1^*$
  or, equivalently

– by 2 fixed points of the two-fold iterate: $f^{(2)}(x_j^*) = f(f(x_j^*)) = x_j^* \; (j = 1, 2)$

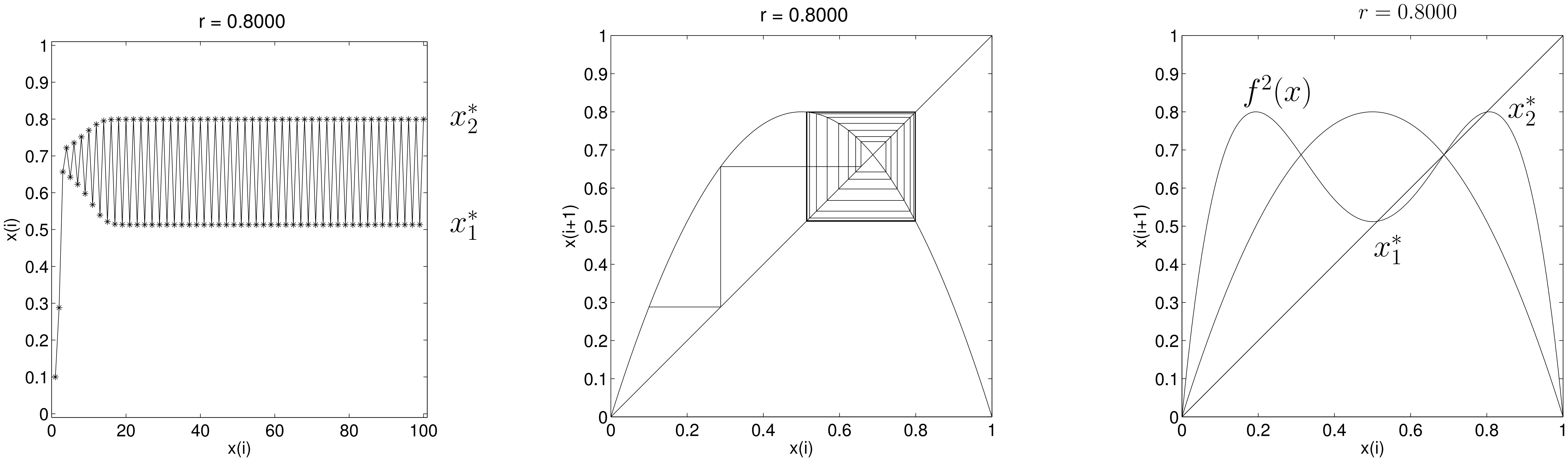


Fig. 5: *Attractor of period 2 for $r = 0.80$, the 2-fold iterate has 2 attractive f.p.*

stability condition is analogous to the previous study for single f.p. of $f(x)$:

$$|f^{(2)'}(x_{1,2}^*)| > 1 \text{ for } r > r_2 = \left(1 + \sqrt{6}\right)/4 \approx 0.8624$$

for $r \gtrsim r_2$ the 2-cycle is replaced by a stable 4-cycle

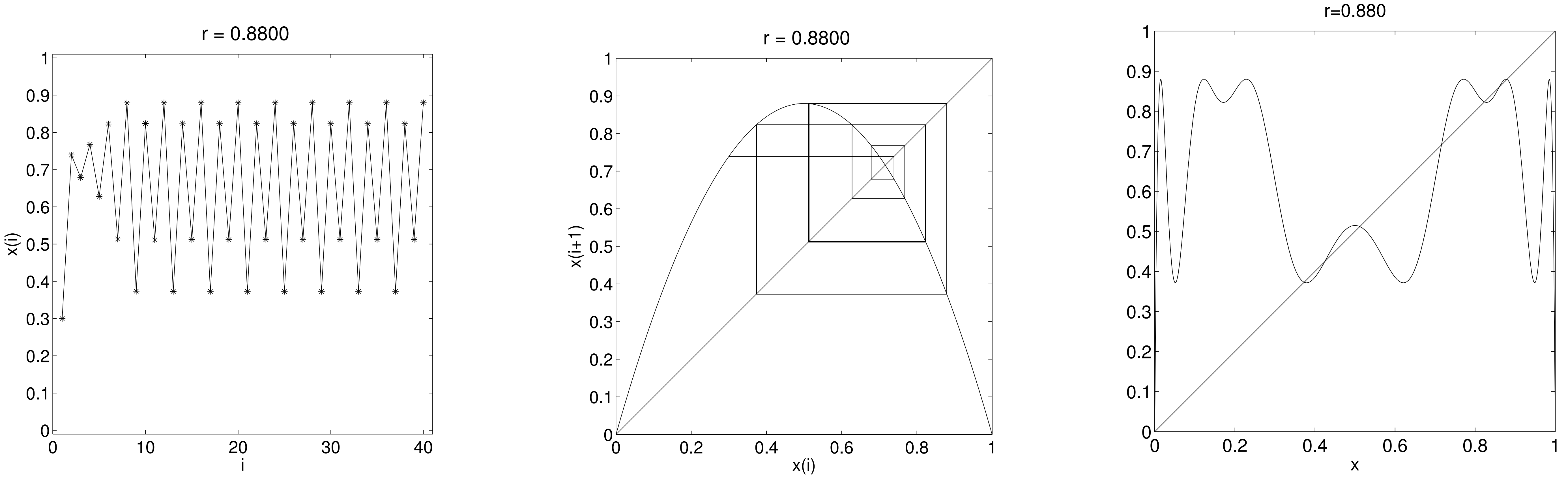


Fig. 6: *Attractor of period 4 for $r = 0.88$, $f^{(4)}$ has 4 attractive and 4 repulsive f.p.*

period doubling scenario:

- at every **bifurcation point** r_k, a 2^{k-1}-cycle becomes unstable and is replaced by a stable 2^k–cycle
- the sequence of r_k approaches for $k \to \infty$ a limiting value $\boldsymbol{r_\infty \approx 0.8925}$

Matlab routine

```
  function rp = rplot(rmin,rmax,rinc,iter,iwaste)

% rplot(rmin,rmax,rinc,iter,iwaste)
% performs  iwaste {800} initial iterations of the
% logistic map, the follwing iter {200} steps are
% plotted versus r in the range (rmin,rmax) with
% increment rinc {0.0001}

  if nargin < 5
     iwaste = 800;
  end
  if nargin < 4
     iter=200;
  end
  if nargin < 3
     rinc=0.0001;
  end

% number of r values
  nofr = floor( (rmax-rmin)/rinc);

  rr=rmin*ones(1,nofr);
  x =  ones(nofr,iter);

  for j=1:nofr
    rr(j)= rmin + j* rinc;
    x(j,1)=logstep(rr(j),0.2,100);
    for i=2:iter
        x(j,i) = logstep(rr(j),x(j,i-1),1);
    end
  end

  plot(rr, x(1:nofr,2:iter),'k.','MarkerSize',0.1)
  axis([rmin rmax 0.0 1.0]);
```

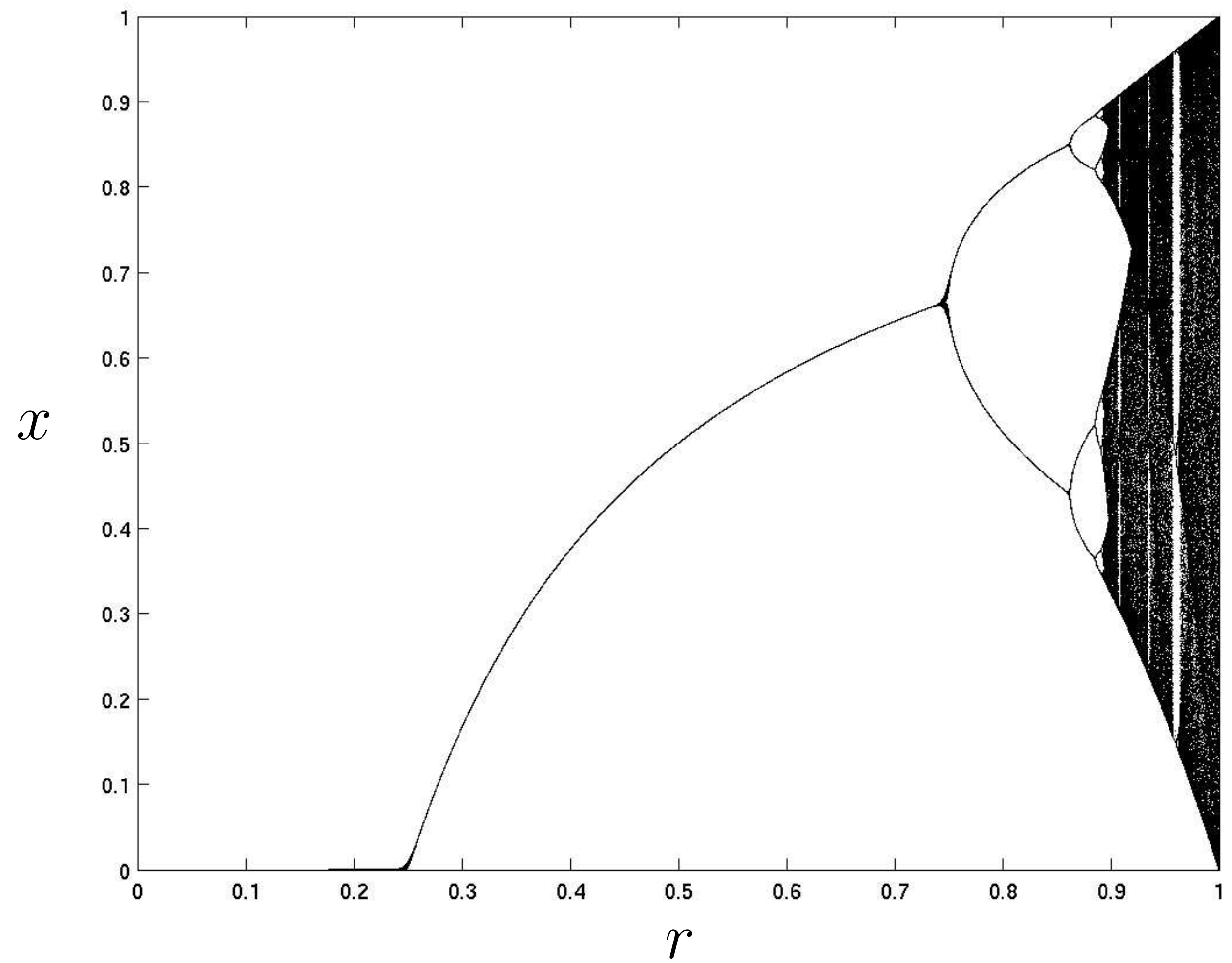


Fig. 7: *The bifurcation diagram, x-values vs. r*

The bifurcation diagram

plot of the x-values (vs. r) that occur in the iteration after an initial phase (here: 800 initial iterations, then 200 values recorded)

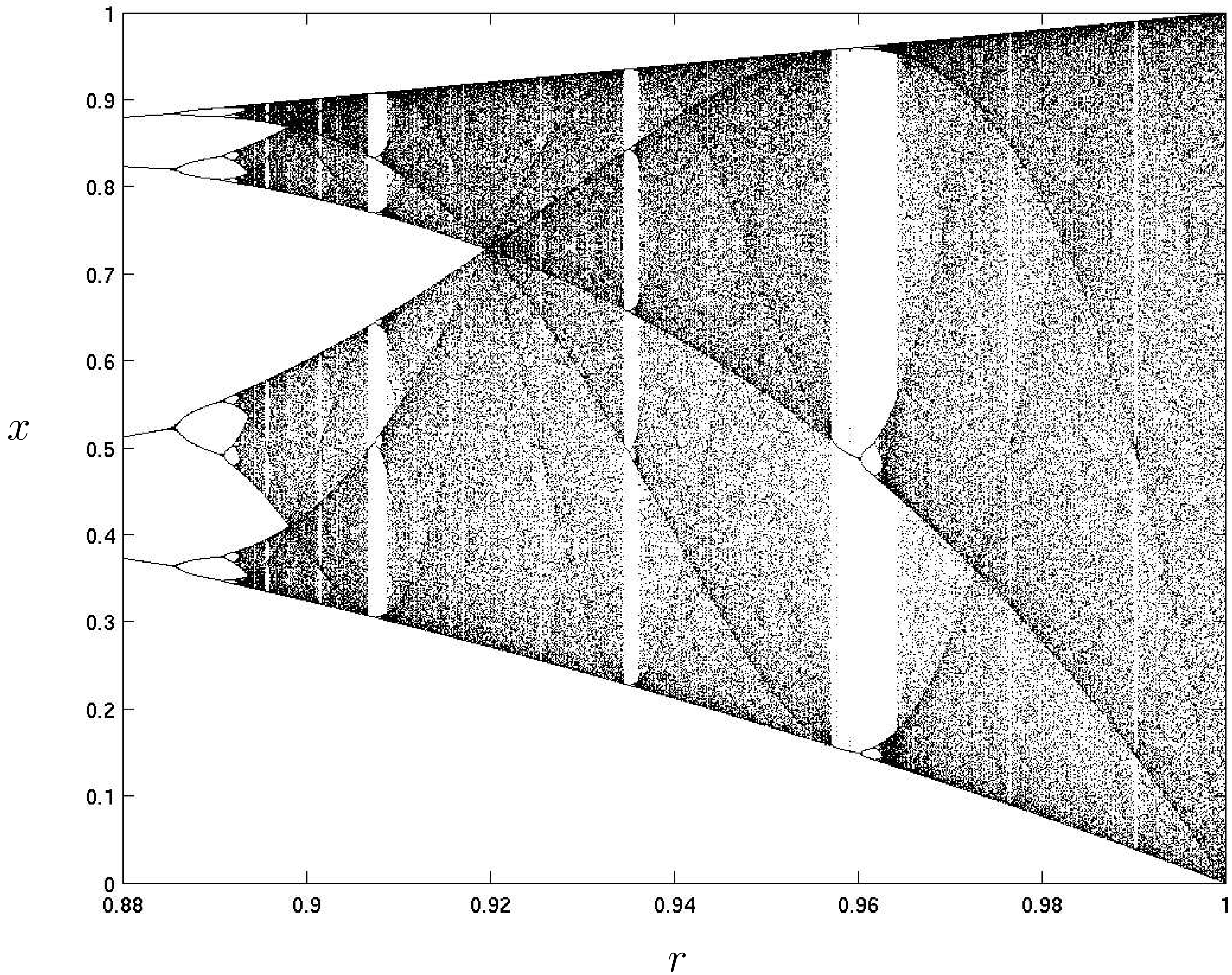


Fig. 8: *The bifurcation diagram in the region of* $r \geq 0.88$

The Feigenbaum constant

Bifurcation points can be identified by searching for fixed points $f^{(\ell)}(x^*) - x^*$ and the r-value where $\left| f^{(\ell)\prime}(x^*) \right| = 1$. Note: this becomes very involved for large ℓ.

An important observation:

the sequence of bifurcation points approaches r_∞ as

$$r_k \approx r_\infty - \frac{c}{\delta^k} \quad \text{for large } \; k$$

with the so called **Feigenbaum constant**

$$\delta = \lim_{k \to \infty} \frac{r_k - r_{k-1}}{r_{k+1} - r_k} \approx 4.669202$$

The importance of the Feigenbaum constant is due to its **universality**.

It occurs in all one-dim. iterations with a *smooth* $f(x)$ *with a single hump* and even in many experimental systems that show **period doubling**.

Matlab routines

```
  function r = channelplot(rr,bins,iter,iwaste,top)
% channelplot(r,bins,iter,iwaste) performs iwaste
% {1000} initial iterations of the log. map with
% parameter r, then generates a histogram for the
% occurence of x-val. in the subsequent iter {10000}
% steps, bins {1000} controls the number of bins,
% top {iter/100} is the maximal occurence displayed

  if nargin < 2
     bins=1000;
  end
  if nargin < 3
     iter=10000;
  end
  if nargin < 4
     iwaste=1000;
  end
  if nargin < 5
     top = iter/100 ;
  end

  x= 0.1 * ones(1,iter);
  x(1) = logstep(rr,0.2,iwaste+1);
  for i=2:iter
      x(i)= logstep(rr,x(i-1),1);
  end
  plot(0,0); axis ([0 1 0 top]); hold on;
  title(['r =' num2str(rr,'%0.5f')],'FontSize',18);
  hist(x,bins);  hold off;
```

```
  function c = channelmovie(rmin,rmax,rinc,dt)

% channelmovie(rmin,rmax,rinc,dt)
% generates a sequence of channel plots for
% r-values between rmin and rmax {1}. r is
% incremented by rinc {0.0001} after dt {1/50} sec.

   if nargin < 4
      dt=1/50;
   end
   if nargin < 3
      rinc = 0.0001;
   end
   if nargin < 2
      rmax = 1;
   end

   number = floor( (rmax-rmin)/rinc )

   for i=1: floor( (rmax-rmin)/rinc)
     r = rmin + i * rinc;
     channelplot(r)
     pause(dt)
   end
```

Period doubling

histograms for the occurence of x-values for increasing r:

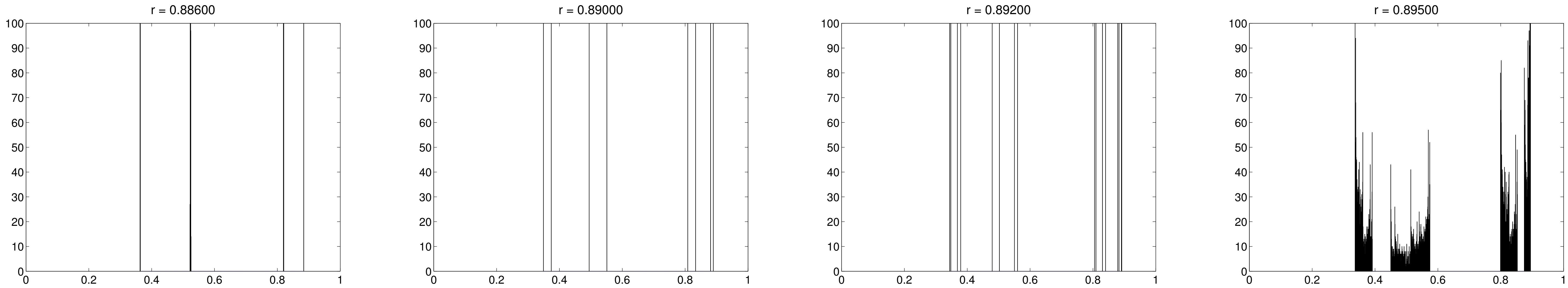
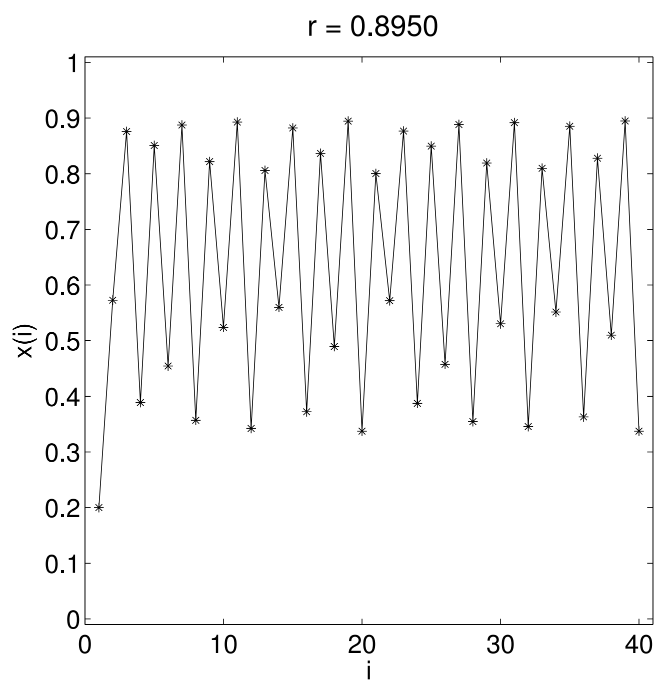
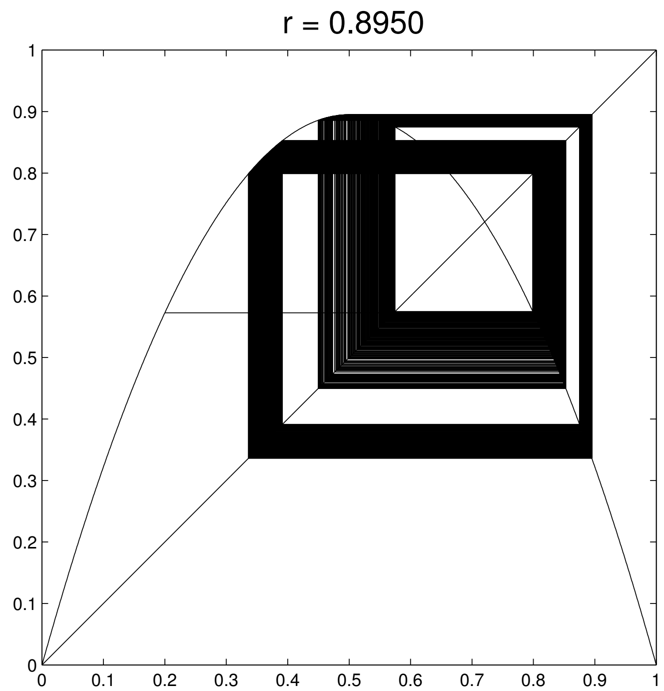


Fig. 9: *Stable attractors with periods 4,8,16 and a chaotic orbit with 4 bands*

The chaotic region

for $r \gtrsim r_\infty$ periodic attractors are replaced by *chaotic orbits* which **fil** (one or several) intervals of x-values by means of a *quasiperiodic sequence* :





Windows of periodic behavior

within the chaotic region one finds *windows* with peridic orbits:

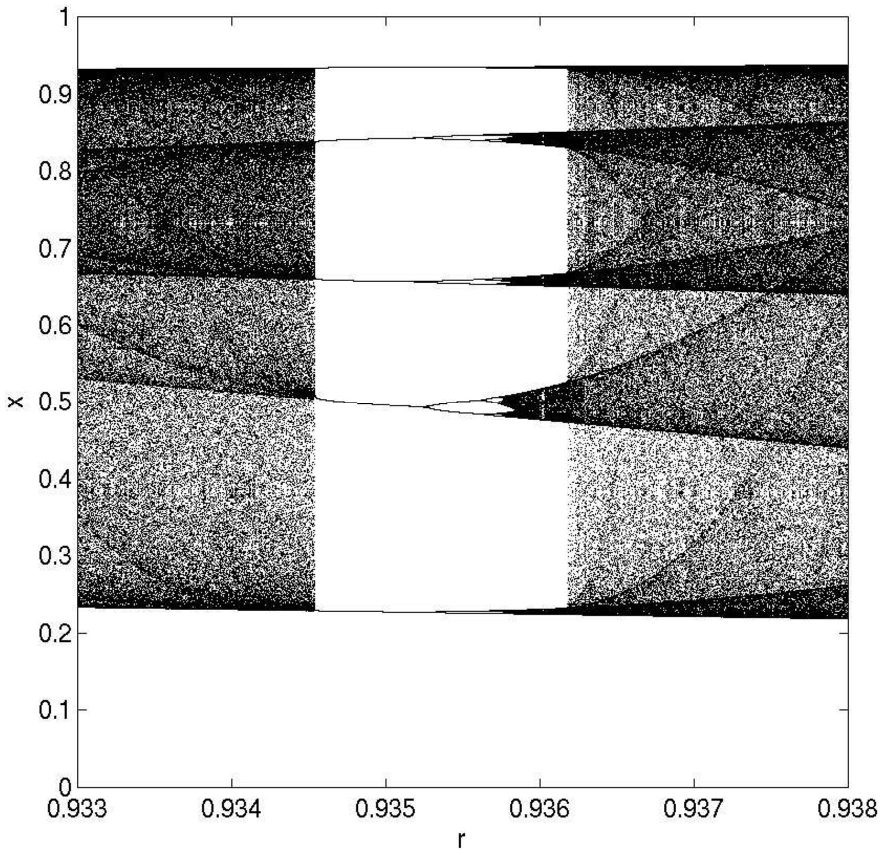


Fig. 11: *A window of period 5 in the vicinity of* $r = 0.935$*. Suggestion: run* `channelmovie(0.933,0.938,0.00001)` *to observe how period doubling leads to chaotic behavior again.*

One can show: the bifurcation diagram contains every integer period, i.e. there are infinitely many windows!

In each of these windows w, period doubling leads to chaotic behavior at a limiting value r_∞^w. Again, the sequence of bifurcation points r_k^w converges to r_∞^w according to the Feigenbaum scenario:

$$r_k^w \approx r_\infty^w - \frac{c^w}{\delta^k}$$

Here, c^w may depend on the actual window, but δ is the Feigenbaum constant.

Iterate, say, 100 times from initial values x_0 and $x_0 + \epsilon$ with, e.g., $\epsilon = 10^{-12}$

– for a **periodic attractor**, e.g. $r = 0.935$:
logstep(0.935,0.2 ,100) yields 0.49617646948447...
logstep(0.935,0.2+ ϵ,100) yields 0.49617646948447...
small initial deviations remain small and vanish!

– for a **chaotic orbit**, e.g. $r = 0.97$:
logstep(0.97,0.2 ,100) yields 0.92484...
logstep(0.97,0.2+ ϵ,100) yields 0.2782...
small initial deviations become very large!

The logistic map

- defines a simple **deterministic** iteration
  (it is not, e.g., a stochastic process)
- becomes **unpredictable** in the chaotic regime
  because of the sensitivity to initial conditions

Note: this is not "just a problem of numerical precision", but, of course, accuracy becomes a problem. In a computer, the x_i are only evaluated to machine precision δ, in general. So, the above results for the chaotic regime are certainly completely "wrong", as already x_1 differs from the true value by $\mathcal{O}(\delta)$.

The Lyapunov exponent quantifies the sensitivity to initial conditions.

Consider the iterations from two slightly different initial values:

$$
\begin{array}{llll}
x_0 & \rightarrow x_1 & \rightarrow x_2 & \ldots \\
x_0 + \epsilon_o & \rightarrow x_1 + \epsilon_1 & \rightarrow x_2 + \epsilon_2 & \ldots
\end{array}
$$

with the sequence of $\epsilon_i, \quad i = 0, 2, \ldots$

After n iterations: $|\epsilon_n| \; = \; \left| f^{(n)}(x_o + \epsilon_o) - f^{(n)}(x_o) \right| \; \equiv \; \epsilon_o \, e^{\lambda n}$

In the limit $\epsilon_o \rightarrow 0$ and $n \rightarrow \infty$, the r.h.s. defines the **Lyapunov exponent** $\boldsymbol{\lambda}$.

Noting that $\displaystyle \lim_{\epsilon_o \rightarrow 0} \; \frac{f^{(n)}(x_o + \epsilon_o) - f^{(n)}(x_o)}{\epsilon_o} \quad \equiv \frac{d\, f^{(n)}(x)}{dx}\bigg|_{x=x_o}$

and using the chain–rule $\frac{d[f(g(x))]}{dx}\big|_{x=x_o} = f'(g(x_o))\, g'(x_o)$

we can write

$$
\lambda \; = = \lim_{n\rightarrow\infty} \; \frac{1}{n} \; \ln \left| \; \frac{df(f(f \ldots f(f(x))\ldots))}{dx}\bigg|_{x=x_o} \; \right| \quad = \quad \lim_{n\rightarrow\infty} \; \frac{1}{n} \sum_{i=0}^{n-1} \ln \, |f'(x_i)|.
$$

We can determine λ from a single sequence of x_n!

Lyapunov exp. obtained from scanning r and performing 1000 iter. for each value:

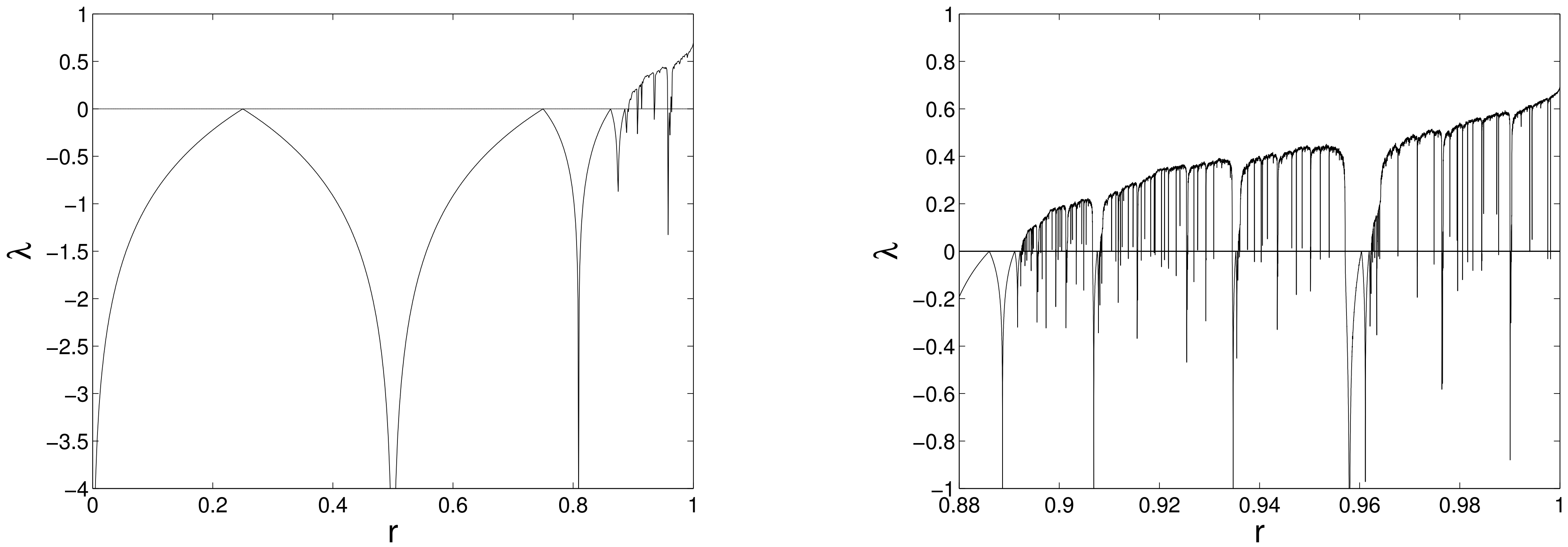


Fig. 12: *Exponent λ vs. r for the entire range (left) and close-up (right) for $r \geq 0.88$. The resolution of the plot is limited by the increment in r (left: 0.001, right:0.0001).*

- $\boldsymbol{\lambda < 0}$: negative exponents indicate **convergent behavior** towards a stable attractor or single fixed point. Small initial deviations vanish as $n \to \infty$.

- $\boldsymbol{\lambda > 0}$: positive exponents indicate **chaotic behavior**. Initial deviations grow exponentially with n and the system is very sensitive to initial deviations.

special cases:

- $\boldsymbol{\lambda = 0}$: zero exponents indicate **marginal stability**.
  Values of r with $\lambda \to 0^-$ mark bifurcation points where an attractor becomes unstable and is replaced by one with double period. In principle we could determine the r_k from the Lyapunov plot in order to estimate the Feigenbaum constant, for instance.

- $\boldsymbol{\lambda \longrightarrow -\infty}$: diverging exponents mark **superstable orbits**
  in which deviations (close to the attractor) vanish extremely fast.
  One finds $\lambda \to -\infty$ for all periodic cycles that contain the value $x = 1/2$, because $f'(1/2) = 0$ and $\ln f'(1/2) \to -\infty$.

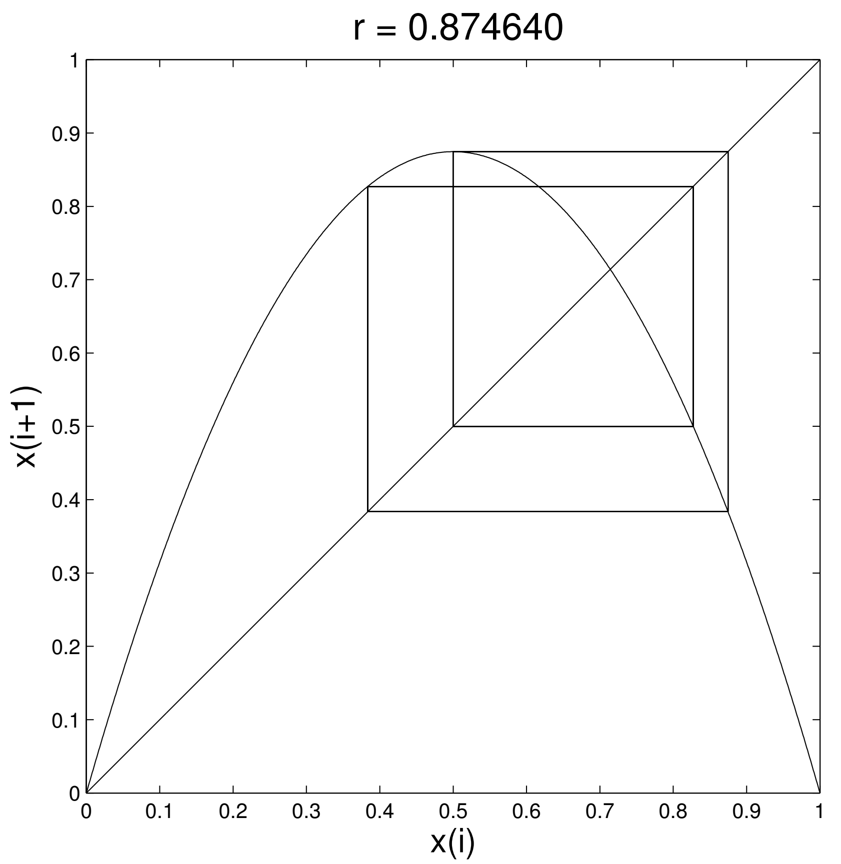


Fig. 13: *Superstable attractor with period four.*

In each interval $[r_k, r_{k+1}]$ of r-values with periodic orbits, there is exactly one value $r = R_k$ which yields a superstable cycle.
Furthermore, the R_k approach r_∞ according to the Feigenbaum scenario. Corresponding statements hold true for all windows of periodic behavior in the chaotic regime.