AC 2008-1079: MATLAB PROGRAMMING FOR VISUALIZING NUMERICAL ITERATIVE DYNAMICS

S.K. Sen, Florida Institute of Technology

Gholam Ali Shaykhian, NASA

MATLAB PROGRAMMING FOR VISUALIZING NUMERICAL ITERATIVE DYNAMICS

Abstract Presented here are the fundamental principles of discrete dynamical system and chaos from the point of view of numerical iterative algorithms. Also included is the visualization of some of such dynamical systems using Matlab programs. Such a visualization has a profound impact on our conceptual clarity and knowledge of the real world scientific and engineering problems.

1. Introduction

The observation that determinism may not imply predictable or regular (or periodic) behaviour influenced many areas of science and engineering. This observation led to the discovery of chaos that has had a significant impact on our conceptual clarity and knowledge of the real-world problems. The purpose of this article is to introduce the fundamental principles of discrete dynamical systems and chaos from the point of view of numerical algorithms and visualizing such dynamical systems using Matlab. This subject throws more light on the chaotic physical phenomena arising in the biochemical reactions, lasers, fluid flow, hurricane, tornedo, and earthquake. The study of chaos or, more generally nonlinear dynamical systems, became a fascinating area since early 1960's (Lorenz 1963; Baker and Gollub 1996; Flake 2001).

The dictionary meaning of the word "chaos" is complete disorder or confusion. Chaos in science and engineering refers to an apparent lack of order in a system that nevertheless obeys certain laws and rules. This understanding of chaos is the same as that of dynamical instability. Deterministic system can produce results which are chaotic and appear to be random. But these are not technically random because the events can be modeled by a nonlinear procedure/formula. The pseudo-random number generator in a computer is an example of such a system. A system which is stable, linear or non-chaotic under certain conditions may degenerate into randomness or unpredictability (may be partial) under other conditions. A pattern may still be discovered in this system on a different level. In *chaos versus randomness*, the common characteristic is unpredictability (at least partial). Observe that absolute randomness does not exist either in material universe/nature or in artificial environment (e.g., in computer) since any outcome has to follow the laws of nature and certain procedures.

Chaos is the irregular and unpredictable time evolution of a nonlinear dynamical system; the system does not repeat its past behaviour even approximately. It occurs in rotating fluids, mechanical oscillators, and some chemical reactions. In spite of its irregularity, chaotic behaviour obeys a deterministic set of equations such as those based on Newton's second law of motion. For example, if we start a a nonchaotic system twice with slightly different initial conditions, the uncertainty (due to, say, measurement errors) leads only to an error in prediction that grows *linearly* with time and the state of the system is known after a short time. However, in a chaotic dynamical system, the error grows *exponentially* with time so that the state of the system is not known after a short time.

Discrete dynamical systems are essentially iterated function systems and hence computers are best suited for this monotonous simple task of simulating such systems and producing beautiful images and visualizing them. Specifically, we will be concerned with the relationship between iterative numerical methods and dynamical systems. We will define the following concepts: function mapping, fixed points (also called attractors or sinks that have integer geometric dimension), orbits and periodicity, points of repulsion (also called a source or a repeller) from where the iteration diverges, and conjugacy among functions. We then explain graphical analysis (also called cob-web analysis due to the appearance of the graph) of the trajectories, their sensitivity to initial conditions, and the chaotic behaviour that result in strange attractors having a nonintegral or fractal geometric dimension. This subject is presently developing fast and is known by three different names such as *Chaotic Numerics, Numerical Dynamics* and *Dynamics of Numerics* and lies in the border of the Turing computable and noncomputable problems (Krishnamurthy 1983). This article is aimed at enabling students to understand the relationship among mathematics, numerical algorithms, and iterative dynamical system so that they can eventually study the fundamental aspects of nonlinear phenomena that pervade all sciences and engineering, in a more detailed manner (Enns and McGuire 1997).

Algorithms and mathematics While studying numerical algorithms (Krishnamurthy and Sen 2001), we are interested in two main aspects, viz., (i) the algorithmic aspect and (ii) the mathematical aspect. The algorithmic aspect provides the dynamics of the iterative scheme in terms of repetition and control, while the mathematical aspect is concerned with the rate of convergence (or speed) to the solution using norms, stability, and also the computational complexity aspects. Typical examples are the Newton's and related gradient methods (Chaps. 1, 3, 11, 12 (Krishnamurthy and Sen 2001)) and QR algorithm (Chap. 6 (Krishnamurthy and Sen 2001)). These iterative procedures are *the most fascinating algorithms* in numerical analysis since these are remarkably efficient although there exist relatively few global convergence results. In fact in many cases, what we can arrive at are sufficient conditions that are often conceptual and are not accessible to numerical practice since these conditions are hard to test in numerical examples. Accordingly, nonconvergence (divergence, oscillation, or stationarity which may be an approximate or different solution due to the finite precision of the machine) arises. This nonconvergence is a difficult open problem in iterative dynamics. Thus the mathematical aspects of convergence are never realistic and the algorithmic dynamics is far beyond the grasp of the mathematical formalism. Therefore, in practice, we need to run the iterative algorithm as a simulation involving trial and error. Thus interactive mathematical software tools, such as Mathematica, Maple, Matlab, become the key tools for understanding the dynamics of iterative algorithms and for visualizing them pictorially with a least effort.

In Section 2, we discuss graphical analysis of iterative dynamics while in section 3, we present Sarkovskii theorem on periodicity. The iterative dynamics of bifurcation and that in higher dimensions including the dynamics of Newton's iterative methods, chaotic numerical dynamics, as well as iterative dynamics for a solution of differential equations are described in section 4 while scetion 5 includes conclusions.

2. Graphical Analysis of Iterative Dynamics

To represent the dynamics of iteration and its analysis we use the two different types of graphs: (i) the topological directed graph with nodes and edges to describe state transitions, and (ii) the function graph with suitable axes to represent the function application and its new values.

Phase Portrait The topological description of an iteration is called the phase portrait. It is a state graph where the iterative steps move the dynamical system from one state to another under each iteration giving rise to an infinite state diagram. For a simple function, phase portraits can aid visualization of the convergence, divergence, and periodicity of an orbit. However, since the phase portrait is essentially topological, we need a more realistic representation through function values. This will be discussed under cobweb analysis.

Examples (i) Consider the phase portrait of $f(x) = x^2$, $f(-1) = 1$, $f(0) = 0$, $f(1) = 1$. For a real function the points in the interval $(-1, 0)$ are mapped into interval $(0, 1)$ and then move towards 0 under iteration. Similarly points which are to the left of -1 are mapped to the right of 1 and then move towards infinity as the iterations proceed.

(ii) Consider $f(x) = -x^3$. Here, 0 is a fixed point and the point 1 goes to -1 and then returns to 1 in the second iteration. Points which are greater than 1 in absolute value oscillate from side to side of zero to another under iteration and grow towards infinity. Points which are less than modulus 1, oscillate and shrink, and ultimately are attracted towards zero. Thus assuming three states -1, 0, 1 along the horizontal axis we can draw a phase portrait. The following Matlab program saved in "minusxcubed", where the initial x is taken as 0.99 will produce the graph (Fig. 1) of $-x^3$ against x using the iterative scheme $x = -x^3$.

%minusxcubed

```
x=.99;for i=1:20, x=-x^3; y(i)=x; end; x1(1)=.99;
for i=2:20, x1(i)=y(i-1); end; plot(x1,y)
```

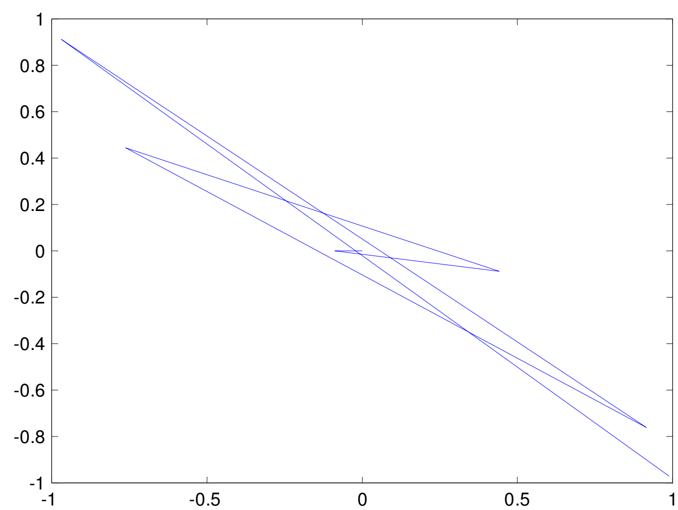

Fig. 1 Graph for $-x^3$ versus x graph for the iterative scheme $x = -x^3$ using the foregoing 10-line Matlab program "minusxcubed"

Replacing the last plot command, viz., **plot(x1,y)**, by **plot (y)** in the foregoing Matlab program, we get the iterates (array) y as in Fig. 2. Observe, from Fig. 2, that the iterates converge to the fixed point 0.

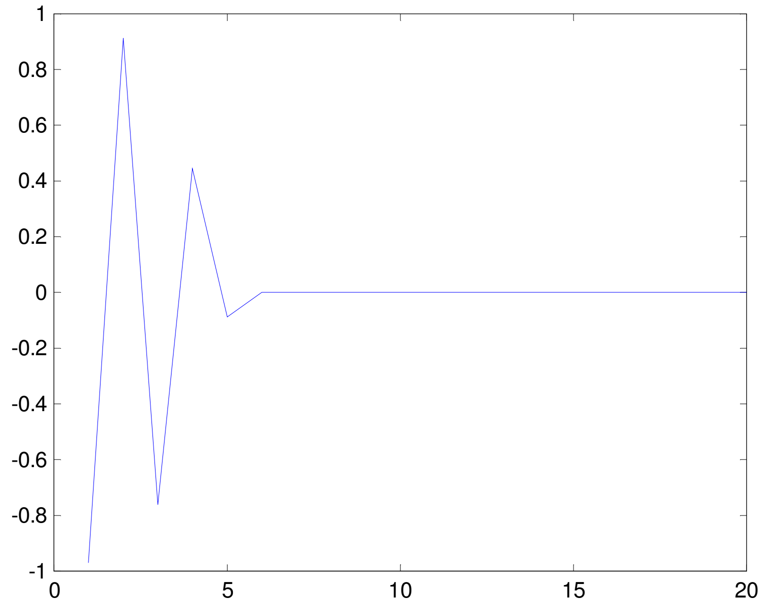

Fig. 2 Graph for the iterates of y obtained from the scheme for $x = -x^3$ using the Matlab program "minusxcubed" with **plot(y)** in place of **plot(x1, y)**

Cobweb analysis The geometric analysis of the dynamics of a function under successive self-applications (or iterations) is called *graphical analysis* or *cobweb*[1] *analysis* due to the physical appearance of the graph obtained. The cobweb analysis of the iteration x(n + 1) = f(x(n)) is carried out as follows:

S.1 First plot the function f(x), along the vertical y axis against x in the horizontal x axis.. This means that we are plotting the iterate x (n) along x axis and x(n+1) along the y axis

S. 2 Draw the y = x or x(n+1) = x(n) diagonal line of slope unity.

S3 Substitute x(0) in f(x) to obtain x(1) = f(x(0)). (f(x(0)) is the intersection of the vertical line from x(0) and f(x) graph.) Now to get x(1) we move horizontally to the diagonal line. The x-coordinate of this intersection is the new x(1). We now use x(1) and repeat the process to find f(x(1)) = x(2) and so on.

The graph so produced will be of three types. In the first type, the orbit converges either monotonically or in an oscillatory manner to a fixed point along the diagonal. In the former case the successive x(k) are in an ascending order. In the latter case the successive iterates occur either side of the fixed point. In the second type, the orbit is periodic with a prime period while in the third type, the orbit has no pattern and is aperiodic. These three types are illustrated below.

Type 1 Consider the iteration $F(x) = 2.9x(1 - x)$ for $0 \le x \le 1$.

Starting from x(0) = 0.1, this iteration converges (rather slowly) to a fixed point at x = 19/29 = 0.6552. The graph of the iterates can be visualized (Fig. 3) using the Matlab commands

x=0.1; for i=1:50, x=2.9*x*(1-x); x1(i)=x; end; plot(x1)

[1] Dictionary meaning is spider's web

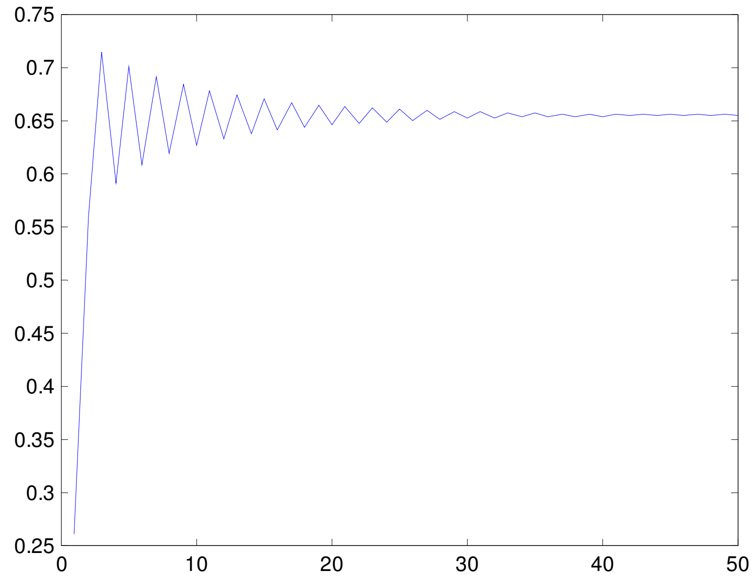

Fig. 3 The graph of for the iterates x = 2.9x(1 − x) starting with x = 0.1 using the foregoing Matlab commands

Type 2 Consider the iteration F(x) = 3.4x(1 − x) for $0 \le x \le 1$. Starting from x(0) = 0.1 this iteration passes very close to the fixed point at x = 12/17 = 0.7059 before approaching the attractive periodic orbit of prime period 2. The graph (Fig. 4) for the iterates can be visualized using the Matlab commands

x=0.1; for i=1:50, x=3.4*x*(1-x); x1(i)=x; end; plot (x1)

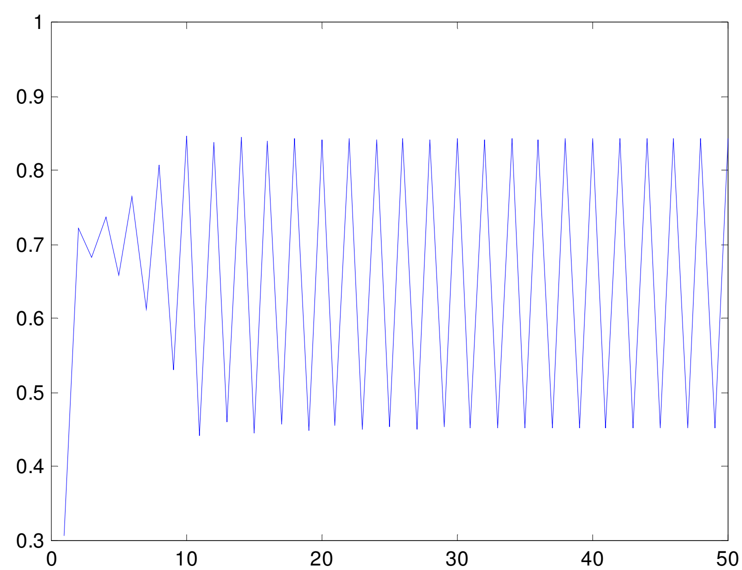

Fig. 4 The graph for the iterates for of x = 3.4x(1 − x) starting with x = 0.1 using the foregoing Matlab commands.

Type 3 Consider the iteration $F(x) = 4x(1 - x)$ for $0 \le x \le 1$. Starting from $x(0) = 0.1$ we do not find any pattern. The graph (Fig. 5) for the iterates can be visualized using the Matlab commands

```
x=0.1; for i=1:50, x=4*x*(1-x); x1(i)=x; end; plot (x1)
```

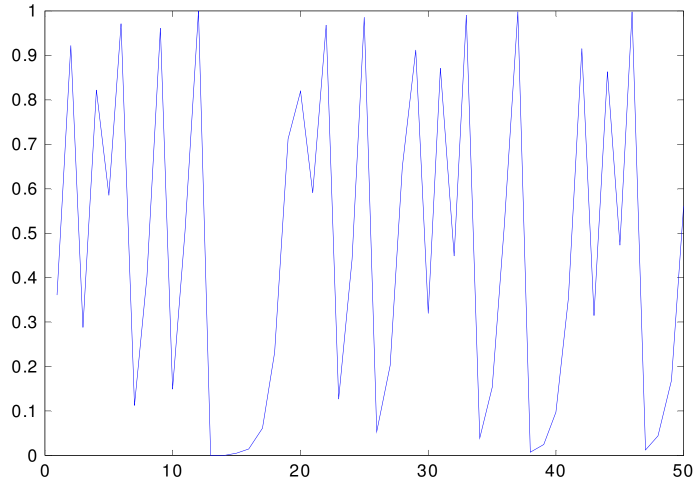

Fig. 5 Graph of for the iterates $x = 4x(1 - x)$ starting with $x = 0.1$ using the foregoing Matlab commands.

Examples (i) We analyse the function $f(x) = x^3$ in the interval $-1 < x < 1$ by plotting successively $(a, f(a))$, $(a, f^2(a))$, . . ., $(a, f^n(a))$ and the diagonal straight line $y = x$ for $a = -1(0.1)1$. Note that $f^n(a)$ approaches zero when a lies in the interval $0 < a < 1$. The graph (Fig. 6) obtained by plotting $f(x)$, $f^2(x)$, $f^3(x)$ for $x = -1(.1)1$ using the Matlab commands

```
x=[-1:.1:1]; z=x; y=x.^3; y1=y.^3; y2=y1.^3;
plot(x, y,'k-', x, y1,'+', x, y2,'o', x, z)
```

saved in "xcubedsuperposed" and executing the program "xcubedsuperposed" is

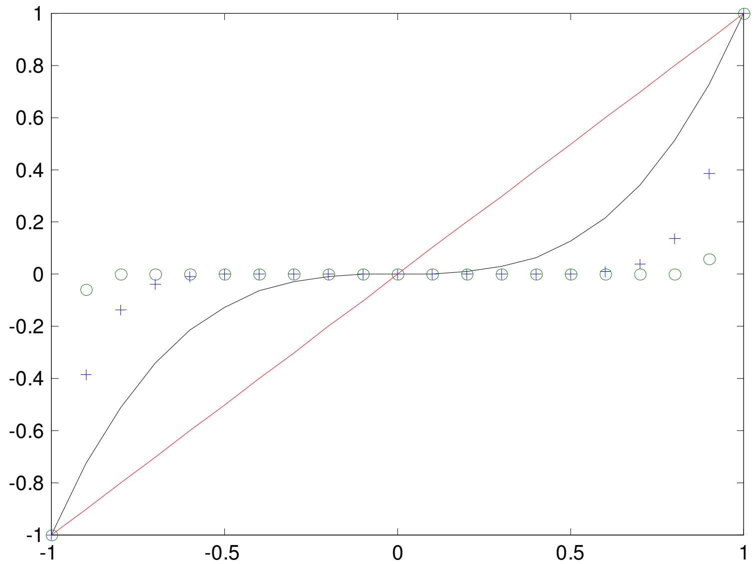

Fig. 6 Graph of $f(x) = x^3$, $f^2(x) = f(f(x)) = x^9$, and $f^3(x) = x^{27}$ for x = −1(.1)1 including diagonal straight line along with the x = y diagonal straight line obtained using the Matlab program "xcubedsuperposed".

Essentially if the iteration is $x(n + 1) = f(x(n))$, where $f(x) = x^3$, then along the horizontal x axis we plot the old x(n) values and in the vertical y axis we plot the new x(n+1) values; then the new x(n+1) value becomes the old value by moving horizontally and repeating the iteration process. This does not lead to a cobweb diagram while $f(x) = -x^3$ does (see Fig. 1).

(ii) Consider $f(x) = -x^{1/3}$. We analyse graphically the dynamics of this iterated function in the interval (0, 1). Note that as n grows, $f^n(x)$ approaches 1 in absolute value and oscillates from one side of the zero to another. Note that −1 and 1 form a periodic cycle with prime period 2. $f^n(x)$ is positive for n even.

(iii*) Discrete iterations* Consider the finite set S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. The discrete iteration F is defined thus: For an element x in S, x is squared and the sum of the digits is formed thus: if the sum is greater than 9, then the digits are again added so that we get a total sum that is an integer between 0 and 9. This iteration F has the following transition graph:

1(cycle), 8 to 1, 2 to 4 to 7, 4 to 7 (cycle), 5 to 7 to 4, 3 to 9 , 6 to 9, 9 (cycle), 0 (cycle)

3. Sarkovskii Theorem on Periodicity

Let f: $R \rightarrow R$ be a continuous function of x The point x is a periodic point of f with a period k if $f^k(x) = x$ and $f^n(x) \neq x$ whenever $0 < n < k$. In other words, x is a periodic point of f with period k, if x is a fixed point of f when f is applied to itself k times. The point x has the prime period p, if $f^p(x) = x$ and $f^n(x) \neq x$ whenever $0 < n < p$. We are interested in the possible period of periodic points of f.

A theorem on periodicity may be stated without proof as follows. If a continuous function of the real numbers has a periodic point of prime period three, then it has periodic points of all prime periods. Before stating Sarkovskii theorem which is a fundamental[2] theorem on periodicity, we need to define the Sarkovskii ordering of integers.

Sarkovskii order of integers The following order of all positive integers defines the Sarkovskii's order; here a <= b indicates that a precedes b in the order:

$3 \le 5 \le 7 \le 9 \le \ldots \le 2\ldots 3 \le 2.5 \le 2\ldots 7 \le 2.9 \le \ldots \le 2^2.3$
$\le 2^2.5 \le 2^2.7 \le 2^2.9 \le \ldots \le 2^n.3 \le 2^n.5 \le 2^n.7 \le 2^n.9 \le \ldots \le 2^4 \le 2^3$
$\le 2^2 \le 2^1 \le 2^0.$

That is, we start with the odd numbers in ascending (increasing) order, then 2 times the odds, 4 times the odds, 8 times the odds, . . ., and at the end we list the powers of 2 in descending (decreasing) order including 2^0.

Sarkovskii theorem The Sarkovskii theorem may be stated as follows. Let f be a continuous function over the reals and f has a periodic point a with period p. If p <= m in the Sarkovskii ordering, then f also has a periodic point of period m.

As a consequence, we see that if f has only finitely many periodic points, then they must all have periods which are powers of two. Furthermore, if there is a periodic point of period three, then there are periodic points of all other periods. Thus, a one dimensional system with a periodic orbit of period three has a periodic orbit of every other positive integer period.

Example Consider h(x) = 3.2x(1 − x) for 0 < x < 1. Here h(x) has fixed points at 0 and near 0.7 i.e., 0.6875). Then h2 = h(h(x)) has a period 2 orbit near 0.5 (i, e., 0.5130) and 0.8 (i.e., 0.7995). Also, h3 = h(h(h(x))) has period 4 points at 0, and near 0.7, 0.5 and 0.8. Note h has no prime period 4 orbit. Therefore h has no orbits other than 2 and 1 since they are the only orbits less than 4 specified in the Sarkovskii order. The commands to obtain *h = h(x), h2 = h(h(x)), h3 = h(h(h(x)))* and the symbolic (analytic) expressions are

```
syms x;
» h=3.2*x*(1-x); g=expand(h); h=collect(g)

h = 16/5*x-16/5*x^2

» h2=3.2*h*(1-h); g=expand(h2); h2=collect(g)

h2 =  256/25*x-5376/125*x^2+8192/125*x^3-4096/125*x^4

» h3=3.2*h2*(1-h2); g=expand(h3); h3=collect(g)

h3 =
4096/125*x-1478656/3125*x^2+47316992/15625*x^3-
806158336/78125*x^4+1577058304/78125*x^5-
1778384896/78125*x^6+1073741824/78125*x^7-268435456/78125*x^8
```

[2] essential or primary

To obtain the fixed points of h(x), we may use the following commands.

»xmh=x-h

»xmh = -11/5*x+16/5*x^2
» fun='-11/5*x+16/5*x^2'; fzero (fun, .1)
Zero found in the interval: [-0.028, 0.19051].
ans = 1.8700e-016
» fzero(fun, .7)
Zero found in the interval: [0.6802, 0.7].
ans = 0.6875

The graph (Fig. 7) of h(x) for x in [0, 1] is

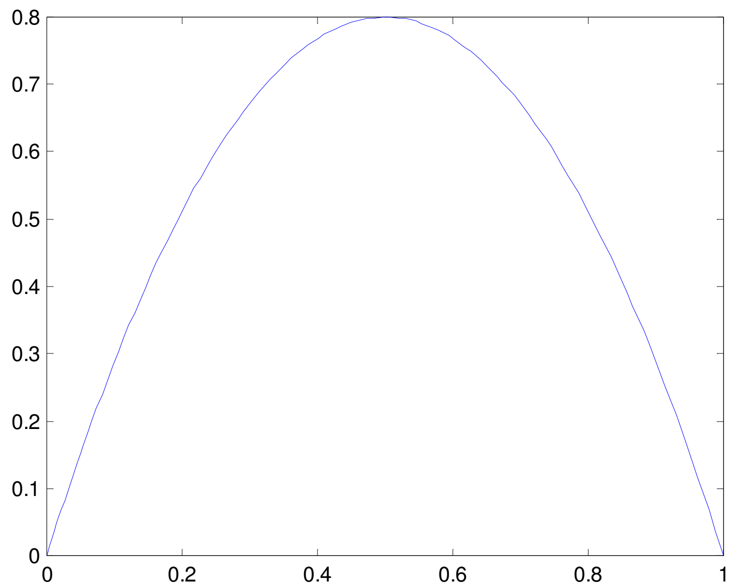

Fig. 7 Graph of *h(x) = 16/5*x-16/5*x^2* for *x* in [0, 1] using the Matlab command **fplot('16/5*x-16/5*x^2', [0,1])**

The graph (Fig.8) for the iteration x = h(x) starting with x = 0.99 is obtained through the Matlab program "sarkovskiiex1" consisting of commands

```
x=.99; for i=1:20, x=3.2*x*(1-x); y(i)=x; end; x1(1)=.99;
  for i=2:20, x1(i)=y(i-1); end; plot(x1,y)
```

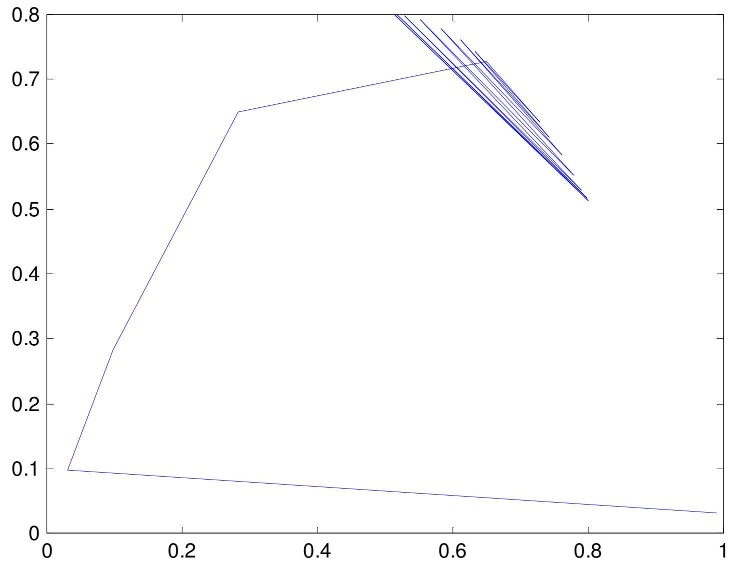

Fig. 8 Graph of the iteration x = 3.2x(1 − x) starting with x = 0.99 using the Matlab program "sarkovskiiex1".

To find the fixed points of $h^2(x) = h(h(x))$, we may use the following commands that include **roots**.

» x-h2

ans = -231/25*x+5376/125*x^2-8192/125*x^3+4096/125*x^4

» c=[+4096/125 -8192/125 +5376/125 -231/25 0];
» roots(c)

```
ans =
        0
   0.7995
   0.6875
   0.5130
```

The graph (Fig. 9) for $h^2(x) = h(h(x))$ for x in [0, 1] is

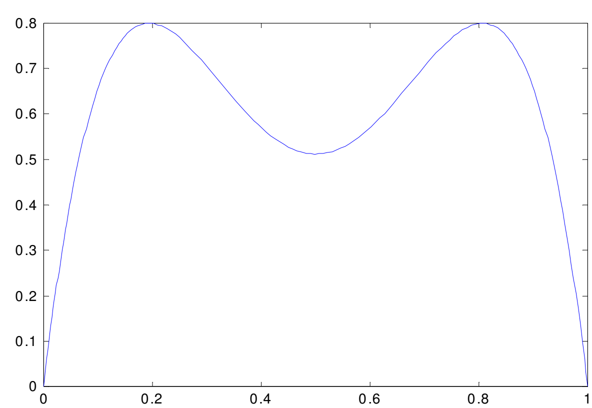

Fig. 9 Graph of h(h(x)) for x in [0, 1] for the using the command **fplot('256/25*x-5376/125*x^2+8192/125*x^3-4096/125*x^4', [0, 1])**

The graph (Fig. 10) for the iteration x = h(h(x)) starting with x = 0.99 obtained through the program "sarkovskiiex11" consisting of the following commands is as in Fig. 10.

```
x=.99;
for i=1:20, x=256/25*x-5376/125*x^2+8192/125*x^3-4096/125*x^4; y(i)=x;
end;
 x1(1)=.99; for i=2:20, x1(i)=y(i-1); end; plot(x1,y)
```

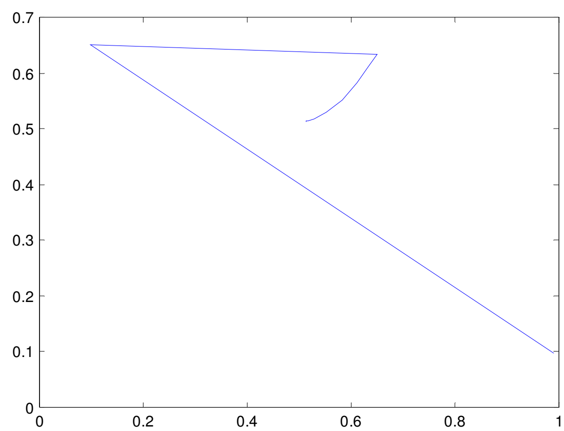

Fig. 10 Graph of the iteration x = $h^2(x)$ starting with x = 0.99 using the Matlab program "sarkovskiiex11".

Sarkovskii's theorem only states that there are cycles of periods and makes no mention whether there are stable cycles of these periods. In reality there must be cycles of all periods, but they are not stable and hence may not be visible on the computer generated graph. This is because computer pictures have a very limited precision to provide a true representation of the real picture.

Remarks (i) The discrete power spectrum is a valuable tool in nonlinear dynamics. The discrete power spectrum is defined by

$$P(n(k)) = |a(k)|^2$$

where n(k), k=1(1)N are the frequencies and a(k) the coefficients in the discrete Fourier transform (DFT) of n(k). Regular motion shows as spikes. A broadband power spectrum with more or less continuous background with a few spikes over is an indication of chaos, though this indication cannot always be taken as the conclusive test.

(ii) We can define a discrete self-correlation

$$C(m) = (1/N)\ \Sigma[x(j)x(j+m)],$$

where x(j) are discrete Fourier transform (DFT) coefficients and the summation is over j = 1 to N. C(m) decays with m and corresponds to a broadband power spectrum.

Example We compute the discrete power spectrum for the logistic map F(x) = rx(1-x) and note its variation as r varies in the region beyond 3 and 4. The following program

r = 3.56; x(1) = .5; for k = 2:1024, x(k) = r * x(k-1)*(1-x(k-1)); end; spectrum(x);

with r = 3.56 and 1024 points x(k) will plot (Fig. 11) the power spectral density using the **spectrum** command,

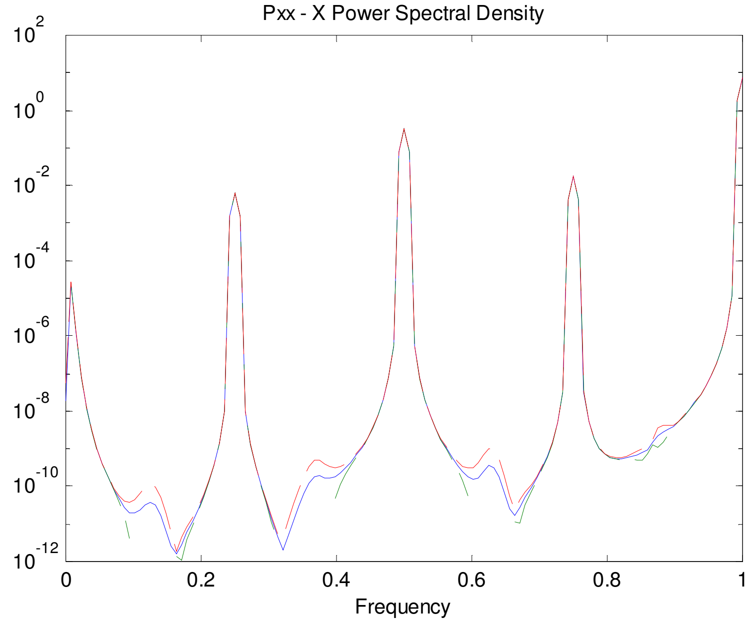

Fig. 11 Graph of the discrete power spectrum for the logistic map F(x) = rx(1-x) where r = 3.56

If we now take r = 3.6 and use the program

r = 3.6; x(1) = .5; for k = 2:1024, x(k) = r * x(k-1)*(1-x(k-1)); end; spectrum(x);

we obtain an entirely different graph (Fig. 12).

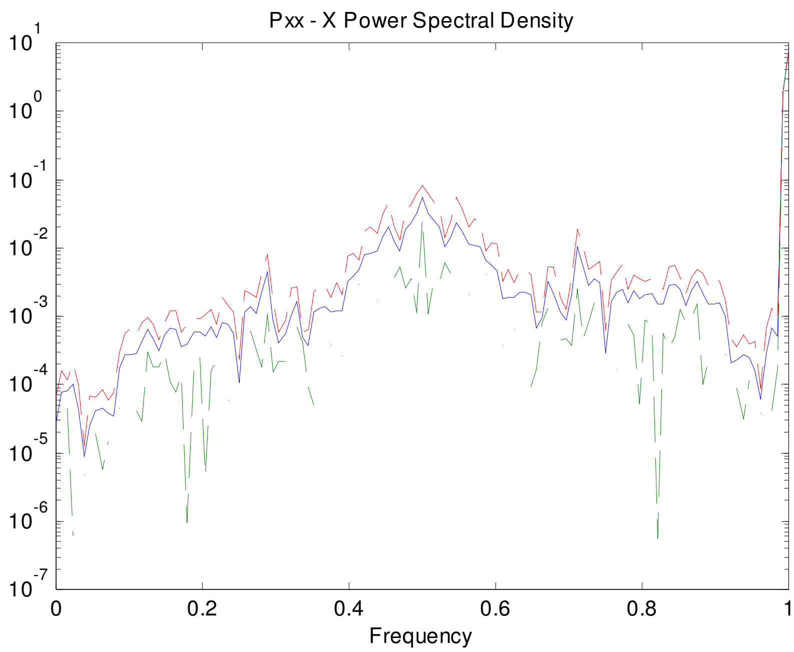

Fig. 12 Graph of the discrete power spectrum for the logistic map F(x) = rx(1-x) where r = 3.6

If we use the commands

r = 3.56; x(1) = .5; for k = 2:1024, x(k) = r * x(k-1)*(1-x(k-1)); end;
spectrum(x,1024,0);

for r =3.56, then we get Fig. 13.

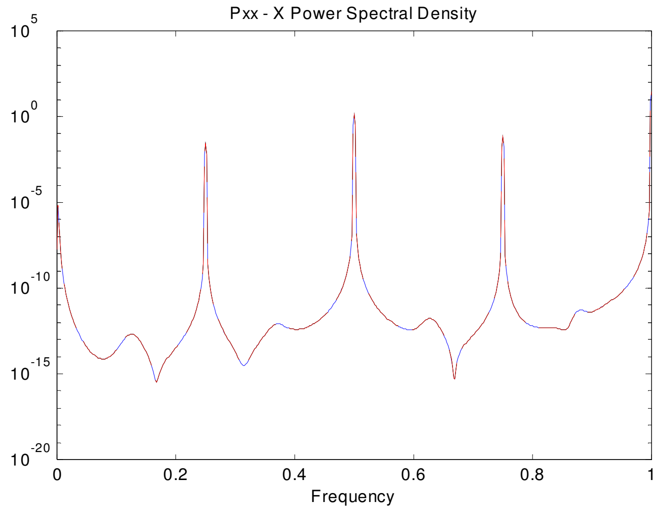

Fig. 13 Graph of the discrete power spectrum for the logistic map F(x) = rx(1-x) where r = 3.56 and the Matlab spectrum command is **spectrum(x,1024,0)**

One may check the **spectrum** commands using the Matlab **help** and observe the differences in various formats of "**spectrum**".

If we now compute the DFT **u** of the 1024 points **x(k)** followed by **z = u .* conj(u);** and plot the points **z** in the log scale using the commands

»r = 3.56; x(1) = .5; for k = 2:1024, x(k) = r * x(k-1)*(1-x(k-1)); end;
» u = fft(x); z = u .* conj(u); plot(log(z));

we obtain Fig. 14. One may compare Figs. 13 and 14 to observe the differences.

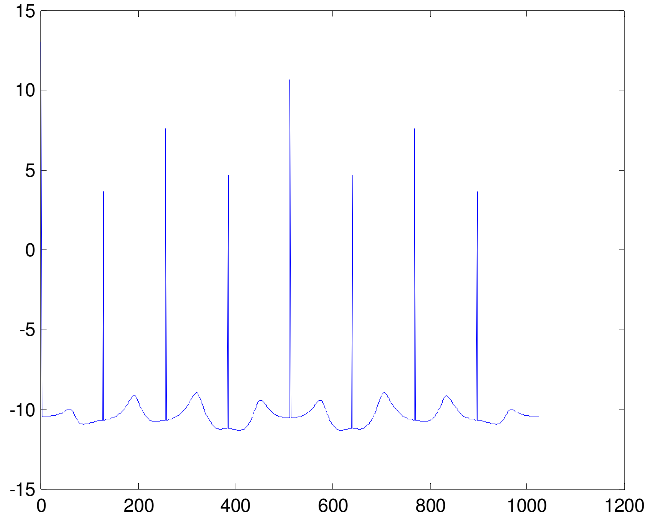

Fig.14 Graph of the discrete power spectrum for the logistic map F(x) = rx(1-x) where r = 3.56 and the Matlab commands are » **u = fft(x); z = u .* conj(u); plot(log(z));**

Iterative dynamics of bifurcation Consider a continuous time system of differential equation, viz.,

dx/dt = f(x, a)

or the discrete iterative system

x(k + 1) = F(x(k), a).

The dynamics of the system whether continuous or discrete is dependent upon the parameter a. In some cases, even a small change in "a" can significantly alter the stability of the trajectory in a qualitative manner, as in an evolutionary tree. Such a change results in a fork in the trajectory for small changes in the parameter. The qualitative change is called "bifurcation" and the value of the parameter at which this change happens is called "bifurcation value". Such bifurcations can therefore be created or annihilated sometimes. The bifurcation can be understood from a change in the nature of the eigenvalues of the corresponding Jacobian[3]. For continuous system we need to check whether the eigenvalues have real parts less than zero for stability or greater than zero for instability. Equivalently, for discrete iterative systems we need to check whether the eigenvalues lie within the unit circle for ensuring stability or whether the eigenvalues lie outside the unit circle causing instability (See Sec. 12). Usually the bifurcation is plotted against the value of the parameter and the resulting graph is called a bifurcation diagram.

In the context of bifurcation, the concept of equilibrium points is important. An equilibrium point is the point that remains unchanged. It is a fixed point in the fixed-point iteration $x_{n+1} = f(x_n)$. For example, an equilibrium point is the fixed point p such that p = f(p). Let the population of tigers living in an inhospitable forest be described by the iterative scheme $x_{n+1} = f(x_n) = 0.9x_n + 40$. Left alone the population would decrease by 10% in each generation (iteration). But there is an immigration of tigers into the forest

[3] Here, by Jacobian, we mean Jacobian matrix. If we have n functions $f_i(x_1, x_2, \ldots, x_n)$ i= 1(1)n, then the n × n matrix whose (i, j)th element is $\partial f_i/\partial x_j$ is the Jacobian. Observe that the determinant of this matrix is the Jacobian determinant which is sometimes referred to as "Jacobian". However, from the context, it will not be difficult to make out whether the term Jacobian means Jacobian matrix or Jacobian determinant.

from the nearby forest at the rate of 40 per generation. If the current population < 400, then 10% decrease will be less than 40. Consequently, the population will increase. On the other hand, if the population > 400, then the population will decrease. If the population = 400, then the population will remain stationary. Thus the point x = 400 is an equilibrium point. We can find all the equilibrium points by solving the equation x = f(x).

Consider the nonlinear logistic[4] model $x = 3x(1 - 0.002x)$. Its two equilibrium points, viz. 0 and 500 are obtained by solving the equation $x - 3x(1 - 0.002x) = 0$. The graph using the Matlab command **fplot('3*(1-.002*x)*x', [0, 500])** is as shown in Fig. 15.

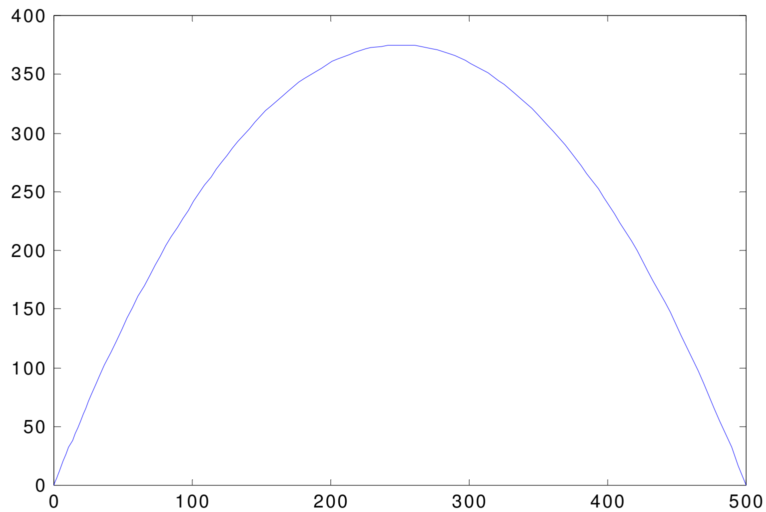

Fig. 15 Graph depicting the equilibrium points 0 and 500 for the nonlinear model $x_{n+1} = 3x_n(1 - 0.002x_n)$ using the Matlab command **fplot('3*(1-.002*x)*x', [0, 500])**

We now illustrate below some special types of bifurcations, namely, saddle node bifurcation, pitchfork bifurcation, period doubling bifurcation, and transcritical bifurcation.

Examples (i) Consider the continuous time system $dx/dt = a - x^2$. For a < 0, this has no equilibrium point. At a ≥ 0 there are two equilibrium points — one at $+\sqrt{a}$ and the other (an unstable one) at $-\sqrt{a}$. These two equilibrium point are obtained by replacing dx/dt by its forward difference form $(x_{k+1} - x_k)/h$ and solving the equation $x = x + ha - hx^2$ derived from the iterative scheme $x_{k+1} = x_k + ha - hx_k^2$. Since h is not 0, the foregoing equation will be $a - x^2 = 0$. To obtain the eigenvalues of the concerned Jacobian (matrix), we write the general continuous time system as $d\mathbf{x}/dt = \mathbf{f}(\mathbf{x})$, where the vector $\mathbf{x} = [x_1 \quad x_2 \ldots x_n]^t$ and the vector $\mathbf{f} = [f_1, \quad f_2 \ldots f_n]^t$. The right-hand side function is time independent. The Jacobian is the n × n matrix $J = [\partial \mathbf{f}_i/\partial x_j]$. For the system $dx/dt = f(x) = a - x^2$, the 1 × 1 Jacobian is $[df/dx] = d(a - x^2)/dx = -2x$ which is $-2\sqrt{a}$ at the equilibrium point $x = \sqrt{a}$ and $2\sqrt{a}$ at the unstable equilibrium point $x = -\sqrt{a}$. Since two equilibrium points are created when a crosses zero, it is a bifurcation with a = 0. This is called a *saddle node bifurcation*. This name is more meaningful in higher dimensions.

[4] The logistic model is the nonlinear one-dimensional model (also may be called a map) given by the difference equation $x_{k+1} = \alpha(1 - x_k)x_k$, $x_k \in [0, 1]$ derived from the corresponding differential equation $dx/dt = \alpha(1 - x)x$. It was originally used to model population development in a fixed (limited) environment. In dictionary, the word "logistic" is the adjective of the word "logistics" meaning "detailed planning and organization of a large, especially military, operation".

(ii) Consider $dx/dt = ax - x^3$. For any value of a the origin is an equilibrium point. Its eigenvalue is equal to a, so it is stable for $a < 0$ and unstable for $a > 0$. For $a > 0$ there are two equilibrium points at $\pm\sqrt{a}$. Both these equilibrium points have eigenvalue $-2a$ so the equilibrium points are stable. A bifurcation occurs at $a = 0$ since at this value, the equilibrium point at the origin changes its stability type and two new equilibrium points are created. Hence it is called a *pitchfork bifurcation*.

(iii) Consider the discrete iterative system $F(x, a) = ax - x^3$ over the reals. Its fixed points are obtained by solving $x = ax - x^3$ or, equivalently, $x(x^2 - a + 1) = 0$. One fixed point is 0 and is independent of a. The other real fixed point exists for $a \leq 1$. When the parameter crosses 1, two more fixed points, viz., $x^* = \sqrt{(a-1)}$ and $x^{**} = -\sqrt{(a-1)}$ appear. At $a = 2$, two periodic orbits appear and these exist for the values $\sqrt{(a+1)}$ and $-\sqrt{(a+1)}$ for every $a > -1$.

(iv) *(Feigenbaum constant)* Consider the logistic equation $F(x, a) = x(k + 1) = 4ax(k)(1 - x(k))$. For $0 \leq a \leq 1$, this map maps the unit interval to itself and we will be interested in the value of a in this range. For this equation, the two fixed points are 0 and $(1 - 1/(4a))$. The first fixed point is stable for $0 \leq a < 0.25$. For the second fixed point, the (slope) eigenvalue (at $1 - 1/(4a)$) is $2 - 4a$. Thus the second fixed point is stable in the region $0.25 < a < 0.75$.

The point $x = (1 - 1/(4a))$ is a sink for every value of a in the interval (0.25, 0.75). At $a = 0.75$ we have $x = 2/3$ and the derivative (dF/dx) at $x = 2/3$ is -1. For $a > 0.75$, instability sets in and a periodic orbit of period 2 is born.

If a is increased further, the period 2 orbit becomes unstable and spawns a stable period 4 closed orbit. This period 4 orbit then spawns a period of eight orbit and so on. Hence this bifurcation is called a *period doubling bifurcation*. This system ultimately leads to chaos and hence is popularly known as “a route to chaos”. The period-doubling bifurcation has another important property.

If a(k) is the bifurcation value for the period 2^k, branching into a $2^{(k+1)}$ period, then limit k tends to infinity, and $[a(k) - a(k-1)]/[a(k+1) - a(k)]$ converges to a value 4.669, called the Feigenbaum constant. This constant is of great importance in complex systems theory. For example, a(1) = 3, a(2) =3.449, a(3) = 3.544, a(4) = 3.564, a(5) = 3.568, a(6) = 3.569.

(v) The function $h(x) = rx(1 - x)$ exhibits another type of bifurcation called *“transcritical bifurcation”***.** When $0 < r < 1$, h has two fixed points, one less than 1 that is repelling and one at zero that is attracting . When $r = 1$ these two fixed points merge at 0 which repels points less than 0 and attracts points greater than zero.

Iterative dynamics in higher dimensions We now consider the iterative dynamics of nonlinear equations in more than one variable. The multivariable case is more difficult than the scalar case for a variety of reasons. These are as follows.

(i) A much wider range of behaviour is possible, and hence the theoretical analysis and the determination of the number of solutions it ha are much more complex.

(ii) It is not possible to devise a safe method that can guarantee the convergence to the correct solution or can determine a narrow interval in which the solution exists.

(iii) Computational overhead increases exponentially with the dimension of the problem (called the *curse of dimensionality*). For example, in the single variable case we were able to find the fixed points analytically by substitution and evaluation. This approach is too complex for a multivariable case, and hard to handle.

We now define some terms that are more relevant and meaningful in the context of multivariable systems. Although we will provide examples of analytical solution to fixed points in simple cases, these are hard to obtain in most cases, and one needs to use numerical techniques. Thus, the examples are only meant for illustrative purposes about the fixed points and their properties.

4. Iteration in Higher Dimension

Iteration in higher dimension The one variable iteration $x = f(x)$ can be extended to the multivariable case if x is considered as a vector of components (elements). Thus a general N-dimensional map is defined by $x(n + 1) = F(x(n))$ with a given/specified starting vector $\mathbf{x}(0)$, where x(n) is a vector in the N-dimensional space. The shape of some hypersphere will depend on the derivatives of the function F(x(n)) with respect to the different components of x(n). The relevant matrix is called the Jacobian matrix or simply Jacobian. For example, if $F = (f(u, v, w), g(u, v, w), h(u, v, w))$ then

$$J = \begin{bmatrix} \partial f/\partial u & \partial f/\partial v & \partial f/\partial w \\ \partial g/\partial u & \partial g/\partial v & \partial g/\partial w \\ \partial h/\partial u & \partial h/\partial v & \partial h/\partial w \end{bmatrix} = [JF]$$

After N iterations, the local shape of the initial hypersphere depends on the matrix

$$J(x(n)) = [[\, JF(x(n))][\, JF(x(n-1))] \ldots [JF(x(0))]],$$

where x is a point (N-vector) in N-dimensional space. In one variable case we said that the convergence rate was determined by $(df/dx)_{x=x^*} = |f'(x^*)|$, where x^* is the solution. If $x = f(x)$, where $f(x) = 0.5x + 4.5/x$, then $f'(x) = .5 - 4.5/x^2$. Let the successive approximations be given as in Table 2. We have here solved the nonlinear equation $x^2 - 9 = 0$ using the Newton scheme (Krishnamurthy and Sen 2001) with the initial approximation $x(0) = 9$.

Table 2. Successive approximations x(i) and corresponding $f'(x(i))$ for $x(i+1) = f(x(i))$.

I	x(i)	$f'(x_i)$
0	9	0.4444
1	5	0.3200
2	3.4	0.1107
3	3.0235	0.0077
4	3.0001	3.3332×10^{-5}
5	3.0000	0

Consider the Jacobian matrix $J(x) = [\partial f_i(x)/\partial x_j]$, where $\partial f_i(x)/\partial x_j$ is the partial derivative of i-th component of f with respect to j-th component of the (independent variable) vector x), See Chapter 11 (Krishnamurthy and Sen 2001) for more details. The analogous condition holds for multivariable case, namely the $r(J(x^*)) < 1$ where r is the spectral radius (Krishnamurthy and Sen 2001) of the Jacobian (gradient) matrix of f evaluated at the point (vector) x^*. As with the scalar case, the smaller the spectral radius, the faster is the convergence rate. In particular, if $r(J(x^*)) = 0$ then the convergence rate is at least quadratic. Newton's method provides for such a possibility.

The fixed-point iteration $\mathbf{x}(k + 1) = F(\mathbf{x}(k))$ converges if the eigenvalues of the Jacobian are less than unity in modulus.

Hyperbolic fixed points Fixed points for which the eigenvalues are not equal to the modulus of unity, i.e., those *which do not lie along the circumference of the unit circle*, are called "hyperbolic fixed points". Hence, the hyperbolic fixed points which have eigenvalues less than 1 in modulus are attracting points (sinks), and those with eigenvalues greater than 1 in modulus are called repelling points (source). The hyperbolic fixed points are structurally stable under perturbations.

Examples (i) Consider $F(x, y) = (y^2 + x/2,\ y/3)$.

Here, the determination of fixed points requires solving two nonlinear two-variable equations, viz., $y^2 + x/2 = x$ and $y/3 = y$. Solving for real values $y = 0$ and hence substitution into the first equation we get $x = 0$. Thus $[0\ \ 0]^t$ is a fixed point. The Jacobian

$$J = \begin{bmatrix} .5 & 2y \\ 0 & 1/3 \end{bmatrix}.$$

The eigenvalues are 0.5 and 1/3 at the fixed point (even otherwise here) and so every orbit of this dynamical system converges to the point $[0\ \ 0]^t$.

Repellers/Source Let $F(x, y) = (2x + y^2 - 2,\ x^2 - x - y^3)$. The fixed points of this system are complicated to be solved analytically; we will only verify that $F(1, 1) = (1, -1)$ and $F(1,-1) = (1, 1)$. This system has a period 2. The two eigenvalues of the derivative of F are computed from the matrix

$$J = \begin{bmatrix} 2 & 2y \\ 2x-1 & -3y^2 \end{bmatrix}$$

The eigenvalues of J at the point (1, 1) are $[-1 \pm \sqrt{33}]/2$, i.e., 2.3723 and -3.3723 while the eigenvalues at the point $(1, -1)$ are $[-1 \pm \sqrt{17}]/2$. i.e., 1.5616 and -2.5616. They are both of modulus greater than 1. Hence the points (1, 1) and $(1, -1)$ are repellers.

(ii) Let $F(x, y) = (-5/3\ x^2 + 2/3\ y + 1, -5/3\ y^2 + 2/3\ x+1)$. We can verify that the orbit at points (0, 0) and (1, 1) is periodic of period 2. The Jacobian is given by

$$J = \begin{bmatrix} -10/3\ x & 2/3 \\ 2/3 & -10/3\ y \end{bmatrix}$$

The eigenvalues of J at x* = (0, 0) are +2/3 and −2/3 while the eigenvalues at x** = (1, 1) are −4 and −8/3.

To find the eigenvalues of the derivative F(F(x)) , we use the extension of the chain rule used in single variable case, namely,

$$d/dx\, F^p(x(i)) = (dF/dx)(x(0))dF/dx(x(1)) \ldots (dF/dx)(x(p-1)),\ \ 0 \le i \le p-1.$$

This takes the form $J(F^p(x(i)) = J(x(0))J(x(1)) \ldots J(x(p-1)),\ 0 \le i \le p-1$. Thus

$$J(F^2(x^*)) = J(0,0)J(1,1) = \begin{bmatrix} 0 & 2/3 \\ 2/3 & 0 \end{bmatrix}\begin{bmatrix} -10/3 & 2/3 \\ 2/3 & -10/3 \end{bmatrix} = \begin{bmatrix} 4/9 & -20/9 \\ -20/9 & 4/9 \end{bmatrix}.$$

The eigenvalues of $J(F^2(x(i))$ are 8/3 and −16/9. Both are greater than 1 in modulus. Thus the orbit is a source.

Saddle A point is said to be a "saddle" if some of the eigenvalues of the Jacobian are of modulus greater than 1 and some of them are less than one in magnitude.

Stable and unstable subspaces Consider the matrix

$$M = \begin{bmatrix} 1/2 & 2 \\ 0 & 3 \end{bmatrix}$$

The eigenvalues and the corresponding eigenvectors are 1/2 , $[1\ \ 0]^t$ and 3, $[4\ \ 5]^t$. Thus any trajectory $[t\ \ 0]^t$ will converge, i.e., will go to zero since $M^n[t\ \ 0]^t = [2^{-n}t\ \ 0]^t$. For this reason the span of (1, 0) is called the stable subspace of M. Similarly, the trajectory starting from any vector of the form $[4t\ \ 5t]^t$ with $t \neq 0$ goes to infinity since $M^n[4t\ \ 5t]^t = [3^n 4t\ \ 3^n 5t]^t$ diverges, Hence the span of (4, 5) is called the unstable sunbspace of M.

Since the vectors u = (4, 5) and v = (1, 0) are a basis, every vector in the 2-D space is a linear combination of the form au + bv, where a and b are arbitrary real numbers. Thus $M^n x = 3^n au + 2^{-n}bv$ and as n tends to infinity the first term goes to infinity while the second term goes to zero.

Dynamics of Newton's iterative method The solution of quadratic equation of the form $ax^2 + bx + c = 0$ is known for many centuries while the solutions of cubic and fourth degree polynomials were discovered only in 16th century. However, it was shown by Galois in the nineteenth century that there is no general method (algorithm) to write the solution of a polynomial of a degree greater than or equal to five, in terms of its coefficient. Such a method based on successive approximations or iterations was developed by Newton and later refined by Raphson. But still a globally convergent purely iterative algorithm has not yet been devised. The Newton method is one of the most prominent algorithms for finding zeros of a function f(x) with real or complex coefficients and is defined by the iterative scheme

$$x(i+1) = x(i) - f(x(i))/f'(x(i)),\ i = 0, 1, 2, \ldots, \text{ till } |x(i+1) - x(i)|/|x(i+1)| \le 0.5 \times 10^{-4} \text{ or } i = 20,$$

where x(0) is a chosen initial approximation. The above iterative sequence may or may not converge to a root of F(x) due to the choice of the initial value x(0) or the nature of the function itself. Observe that the equation F(x) = 0 is written as x = f(x).

An alternative approach to solve the equation f(x)=0 is to use a relaxed (or damped) Newton's method with a relaxation parameter β defined by $x(i+1) = x(i) - \beta\{f(x(i))/f'(x(i))\}$ so that β =1 corresponds to the original Newton method. By making a judicious choice of the relaxation parameter β, the relaxed Newton method is helpful in some cases to speed up the convergence.

Why study Newton method The Newton method and its variants play a fundamental role in numerical methods needed for solving nonlinear systems of equations. Study of its relationship to dynamical systems is carried out to understand its convergence. In particular, we want to classify what are "good" starting points and what are "bad " starting points for Newton methods and also avoid attractive cycles or periodic points other than the zeros.

Also the paradigm $P(Z) = Z^2 + C$ is a very valuable tool to understand the Mandelbrot set, Julia set and Fatou dust and a chaotic dynamical system arising in deterministic computations. Further the extension of the Newton method to the continuous case leads to Euler type differential equations and hence is of importance in studying the stability of differential equations using discrete or finite difference methods of solution and the resulting chaotic dynamics (Peitgen 1989). There are many interesting and deep problems connected with the Newton method, whose solutions will have an impact on practical computing, even though their methodology goes much beyond traditional numerical mathematics. This research is to a large extent experimental in nature, since a mathematical approach is very hard and fortunately interactive software systems on supercomputing systems provide an opportunity to understand them.

Newton's method in higher dimension The Newton method can be extended to n variable case where the iterative scheme of the n-vector **x** with n functional components $f_1(x_1, x_2, \ldots, x_n)$, $f_2(x_1, x_2, \ldots, x_n)$, . . ., $f_n(x_1, x_2, \ldots, x_n)$ is

$$\mathbf{x}(k+1) = G(\mathbf{x}(k)) = \mathbf{x}(k) - f(\mathbf{x}(k))/f'(\mathbf{x}(k)) = x(k) - J^{-1}(\mathbf{x}(k))f(\mathbf{x}(k)),\ k = 0, 1, 2, \ldots, \text{ till}$$
$$\|x(k + 1) - x(k)\|/\|x(k + 1)\| \le 0.5 \times 10^{-4} \text{ or } k = 20, \text{ where } x_0 \text{ is a chosen initial approximation (vector)}$$

The (i, j)th element of the Jacobian matrix J is $\partial f_i/\partial x_j$. There is no need to compute the inverse of J(x(k)) to obtain x(k + 1) although the foregoing scheme is a literal generalization of the Newton scheme in one variable. An iterative scheme needing more than 20 iterations is considered not very desirable. That is why often we do not allow more than k = 20 iterations. The maximum number of iterations permitted is a check not only on slow convergent sequence but also on divergent/oscillatory sequence of iterates. A scheme needing around 6 iterations to obtain about 6 significant digit accuracy is desirable.

Although we could explicitly invert J(x(k)) and obtain x(k + 1), we avoid inversion by solving the linear system, i.e., by finding the unknown n-vector $\Delta x(k) = x(k + 1) - x(k)$ and then computing $x(k + 1) = x(k) + \Delta x(k)$ from the iterative scheme $J(x(k))\Delta x(k) = - f(x(k))$, In this sense, the Newton method replaces a system of nonlinear equations with a system of linear equations, but since the solutions of the two systems are not identical, in general, we need to iterate until a required accuracy is gotten.

Example Consider f(x, y) = (x + 2y – 2, $x^2 + 4y^2 - 4$); its Jacobian matrix is

$$J = \begin{bmatrix} 1 & 2 \\ 2x & 8y \end{bmatrix}$$

Observe that the vector x has two components x and y. Although the same letter x has been used both for vector x and for scalar variable x, the clarity is not lost in the context. If we take x(0) = $[1 \quad 2]^t$, then f(x(0)) = $[3 \quad 13]^t$ and

$$J(x(0)) = \begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix}$$

Solving the system, we obtain x(1) = $[-.83 \quad 1.42]^t$; f(x(1)) = $[0, \ 4.72]^t$;

$$J(x(1)) = \begin{bmatrix} 1 & 2 \\ 1.67 & 11.3 \end{bmatrix}$$

Solving, we get x(2) = $[-.19 \quad 1.10]^t$. These iterations converge to $[0 \quad 1]^t$.

Chaotic numerical dynamics We have provided some idea about *chaos* in Sec. 1. The word *chaos* came from Greek, where it represents the vacant, unfathomable space from which everything arose. In the Olympian myth Gaea sprang from Chaos and became the mother of all things**.** In common usage "chaos" stands for great confusion or disorder. In mathematics, however, "chaos" is not a confusion or disorder but a new kind of emergent property that has a very complex structure within itself. We shall again define and picture "chaos" in the function set-up. For defining a chaotic function we need some additional concepts. We now inrtroduce the following terms.

Let X be a set and d be a metric defined on that set. We then have the following definitions.

Open Subset A subset U of X is open if for each x in U there exists an ε > 0 such that d(x, y) < ε implies y is in U.

Neighbourhood Let ε > 0 and x be in X. The set N (ε, x) = {y in X | d(x, y) < ε} is called an ε neighbourhood or a neighbourhood of x.

Convergence Let x(1), x(2), . . ., x(n) be a sequence of elements of the set X. The sequence converges to x if, for each ε > 0, there exists an integer N such that if k ≥ N then d(x, x(k)) < ε.

Accumulation Point Let S be a subset of X. Then the point x in X is an accumulation point (or a limit point) of S, if every neighbourhood of x contains an element of S which is distinct from x.

Closed Subset A subset of X is closed if it contains all of accumulation points of X.

Dense Subset Let A be a subset of B. Then A is dense in B if every point of B is an accumulation point of A, a point of A or both at the same time. In other words, if A is dense in B and x is in B, then every neighbourhood of x contains an element of A.

Continuity If Y is a set and d′ is a metric on Y then the function f: X → Y is continuous at the point x(0) in X if, for every $\varepsilon > 0$, there exists a $\delta > 0$ such that if x is in X and $d(x(0), x) < \delta$, then $d'(f(x(0), f(x))) < \varepsilon$. A function is continuous if it is continuous at each point of its domain. Note that we use a metric d to measure the distance between functions.

Sensitive Dependence Let D be a metric space with metric d. Then the function f: D → D exhibits sensitive dependence on initial conditions if there exists a $\delta > 0$ such that for all $\varepsilon > 0$ there is a y in D and a natural number n such that $d(x, y) < \varepsilon$ and $d(f^n(x), f^n(y)) > \delta$.

The sensitive dependence implies that if the functions are applied many times and iterated, then any small error in initial conditions may result in very large differences between the predicted behaviour and the actual behaviour of the dynamical system.

Topologically Transitive For all open sets U and V in S, there is an x in U and a natural number n such that $f^n(x)$ is in V. In other words, a topologically transitive map has points which eventually move under iteration from one small neighbourhood to any other. Consequently the system cannot be broken up into disjoint open sets which are invariant under that map.

Chaotic Function Let D be a metric space. Then the function f: D → D is chaotic if (i) the periodic points of f are dense in D, (ii) f is topologically transitive, and (iii) f exhibits sensitive dependence on initial conditions.

Examples (i) The simplest chaotic map is obtained by a left shift operation that results in losing the leftmost digit d(0) in the sequence of digits {d(0), d(1), d(2), . . ., d(n) . . .}; this operation is also called *Bernoulli shift*. It is defined by using a shift operator S thus:

S{d(0), d(1), d(2), . . ., d(n) . . .} = { d(1), d(2), . . ., d(n) . . .}.

This map is continuous, the set of its periodic points is dense, it is topologically transitive and it exhibits sensitive dependence on initial conditions. Thus it is a chaotic map.

(ii) Consider $f(x)= x^2 + 1$; then $N(x) = x - (f(x)/f'(x)) = (x^2 - 1)/(2x)$ if we set r = 1 in the relaxed Newton's method. To get an idea about this iteration we carry out the graphical analysis for N(x) at the initial point x(0) = 0.1 with 100 iterations. This resulting graph (Fig. 16) typically illustrates "chaotic dynamics" defined above. It has periodic points that are dense over the reals, topologically transitive, and exhibits sensitive dependence on initial conditions. This result is proved by showing that N(x) is topologically conjugate to the piecewise linear map g(x) = 2x(mod 1); see Holmgren (1994). The program called "chaoticdynamicsnewton" that generates the graph (Fig.16) is

```
x =.1;x1(1)=x; for n=0:100; x=(x^2-1)/(2*x); y(n+1)=x;x1(n+2)=x;end;
for i=1:100; x2(i)=x1(i); y1(i)= y(i); end; plot (x2, y1)
```

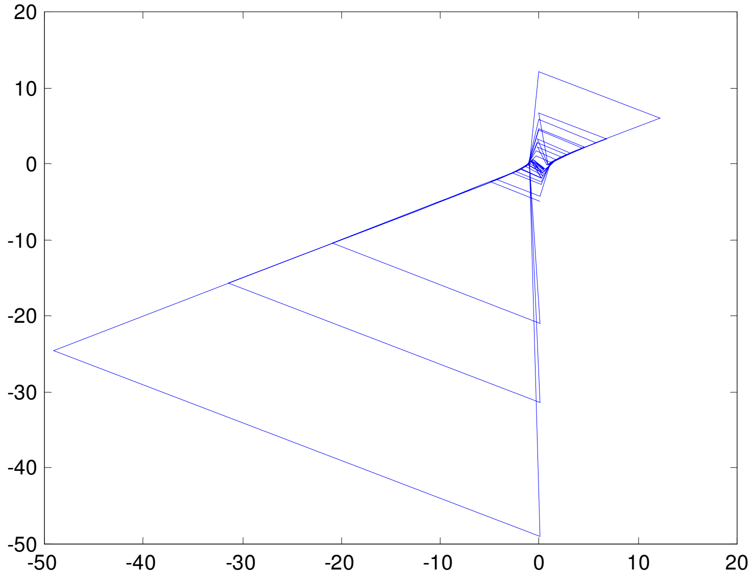

Fig. 16 The graphical analysis of $x = N(x) = (x^2 - 1)/(2x)$ using the program "chaoticdynamicsnewton"

(iii) *Newton's Method over the complex plane: Computer exploration* The Newton method for the complex polynomial f(z) is defined by z = N(z), where $N(z) = z - f(z)/f'(z)$. Let us consider the computation of the nth root of unity, i.e., let us solve the polynomial equation $z^n - 1 = 0$. As a particular case, let $f(z) = z^3 - 1$. Then $N(z) = (2z^3 + 1)/(3z^2)$. This has attracting fixed points at 1, $\omega = \exp(2\pi i/3)$, and $\omega^2 = \exp(4\pi i/3)$.

Select a grid of points covering the square region whose corners are at 2 + 2i, 2 − 2i, −2 − 2i, and −2 + 2i. We calculate the value of the 10th iterate of each these points under iteration of z = N(z) and colour the points as follows.

(a) If the distance from the10th iterate to 1 is less than 1/4, then we assume the point is in the stable set of 1 and colour it blue.

(b) If the distance from the 10th iterate to $\exp(2\pi i/3)$ is less than 1/4,then we assume that the point is in the stable set of $\exp(2\pi i/3)$ and colour it green.

(c) Finally, if the distance from the 10 th iterate to $\exp(4\pi i/3)$ is less than 1/4, then we assume the point is in the stable set of $\exp(4\pi i/3)$ and colour it red.

(d) A point which is not within 1/4 of one of the roots after 100 iterations of N are left uncolored.

With a slight modification of the foregoing logic and using the commands **meshgrid**, **angle**, **image**, and **colormap** (Matlab) commands, we have written a general Matlab program to show the n regions for n zeros of the polynomial $z^n - 1$. The concerned program named 'newtonnthrootof1" for a 251 by 251 grid, is shown in Fig. 17.

%newtonnthrootof1

```
N = 251; rx = linspace( -2, 2, N ); ry = linspace( -2, 2, N );
[x, y] = meshgrid(rx, ry); p = 1; n = 3;% implies cube root of unity
angstep = 2 * pi / n;
for k = 1:N
   for j = 1:N
      z1=rx(k) + i*ry(j); conv = 1;
      for m = 1: 10
         if( abs( z1 ) < 0.00001 ), conv = 0; break; end
```

```
            if( abs(z1) > 200.0 ), conv = 0; break; end
            z1 = ((n-1) * z1^n + 1)/(n*z1^(n-1));
        end;
        if( conv < 1 ), img( 1 + N -j, k ) = 8; else ang = angle(z1);
          if( ang < 0.0 ) ang = ang + 2 * pi; end
          img( j, k ) = 1 + mod( round( 0.1 + ang / angstep ), n );
        end
    end
end
image( rx, ry, img );
%H=image(rx, ry, img );H2=get(H,'Parent' );set(H2,'YDir', 'normal');
colormap( [ 0 0 1; 0 1 0; 1 0 0; 1 1 0; 0 1 1; 1 0 1; 1 1 1; 0 0 0] );
```

Observe that having taken n = 3, we obtain the beautiful graph (Fig. 17) having three regions — one region defining the stable set of 1, the second one containing the stable set of ω while the third one consisting of the stable set of ω^2.

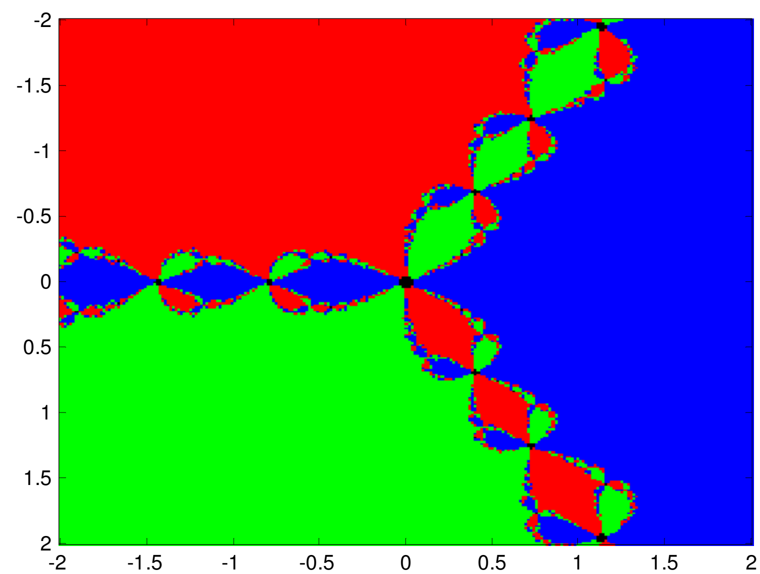

Fig. 17 Graph for the Newton iteration $z = (2z^3 + 1)/(3z^2)$ produced taking initial approximation as each of the 251 × 251 grid points using the Matlab program "newtonnthrootof1"

If we replace the 7^{th} line of the foregoing Matlab program, viz., "`n = 3;% implies cube root of unity`" by "`n = 5;% implies fifth root of unity`", then we obtain the graph (Fig. 18) for the Newton iteration $z = ((n - 1)z^n + 1)/(nz^{n-1})$, where n = 5, with five distinct regions for the five roots.

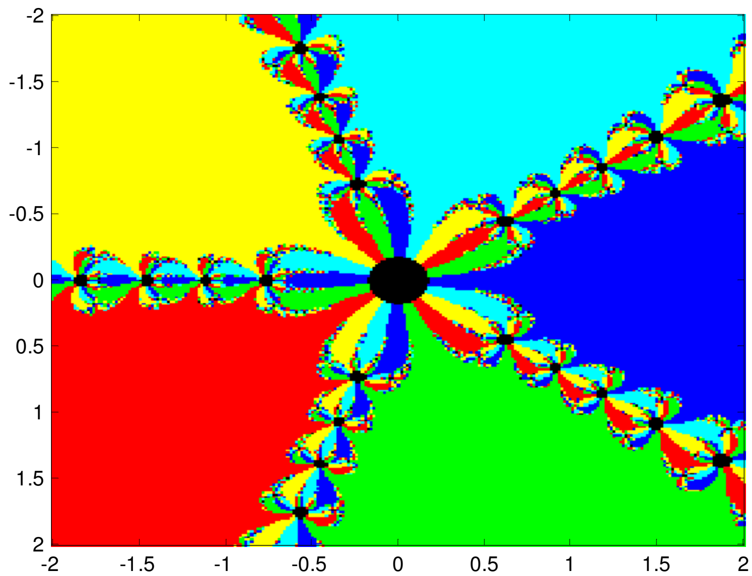

Fig. 18 Graph for the Newton iteration $z = (4z^5 + 1)/(5z^4)$ produced taking initial approximation as each of the 251 × 251 grid points using the Matlab program "newtonnthrootof1"

Observe that the values along the y-axis are in the reverse order in Figs, 17 and 18. This is because the grid point (1, 1) starts from top left and not from bottom; the point (1, 1) corresponds to the smallest value of the axis x and that of the axis y. If we want the values along the y-axis in the normal order, then insert **%** just before the **image** command in the last but two lines and remove **%** from the last but one line of the foregoing Matlab program.

Iterative dynamics of solving differential equations Differential equations have become the language in which many problems in science, economics, and evolutionary growth are encoded. However, solution of such a continuous system through a digital computer requires that the system is modelled by a discrete time system. In this process we are not using a differential equation, but only a simulation of it in discrete time by using different mapping strategies. In such a case, the usefulness of the resulting solution depends not only on the size of the errors, but also the manner in which these errors get magnified as we proceed towards the solution. Therefore, error growth and stability become key issues. In particular we will be interested to find out how the stability of the points of equilibrium are affected in the solution process, in much the same way as we looked for fixed points and their stability in algebraic equations. This is explained below.

Consider the first order linear differential equation $x' = f(x, t)$ with $x(t = 0) = x_0$. All the algorithms to compute the approximate solution of this equation by generating a sequence of points x(0), x(1), x(2), . . . at times t(0), t(1), t(2), . . . respectively. Usually the time steps are uniformly spaced with the step size $h > 0$, that is $t(k+1) = t(0) + kh$ for $k = 0, 1, 2, . . .$ We then approximate the derivative by $x'(t(k)) = [x(k+1) - x(k)]/h$. We can now write the first order differential equation by $x(k+1) = x(k) + hf(x(k),t(k))$ which is the well-known forward Euler discretization algorithm. We assume that as h tends to zero the integration is more accurate. The forward Euler algorithm is called "explicit" since it computes x(k+1) explicitly.

Another type of Euler algorithm is the backward Euler algorithm. Here we approximate the derivative by $x'(t(k)) = [x(k) - x(k - 1)]/h$ and $x(k + 1) = x(k) +$

hf(x(k+1), t(k+1)). This is an implicit algorithm since x(k + 1) is a function of itself (as in a feed-back system). The trapezoidal algorithm $x(k+1) = x(k) + (h/2)[f(x(k), t(k)) + f(x(k + 1), t(k + 1))]$ is a combination of both the implicit and the explicit algorithms taking the average.

For m-step integration we assume that the previous m points lie on the trajectory. However, if these previous points were calculated by approximation they will not lie exactly on the trajectory but somewhere near by. Thus if there are round-off or truncation errors they propagate and can lead to instability leading to unbounded values. This situation is analogous to the unstable fixed point. If the initial condition lies exactly on the equilibrium point, the trajectory should stay there always. The stability or instability of the fixed point determines the stability or instability of the integration algorithm. Thus numerical integration is related to iterative dynamics.

For example, consider the Euler discretization with $f(x) = \lambda x$. Then for the Euler forward algorithm we get $x(k+1) = x(k) + h\lambda x(k) = (1+ h\lambda)x(k)$. The factor $(1+h\lambda)$ is the characteristic multiplier; its role is similar to the role of an eigenvalue. The Euler algorithm is then stable if $|1+ h\lambda| < 1$. This region is called the region of stability of the numerical integration algorithm.

On the other hand for the backward Euler algorithm we have $x(k + 1) = x(k) + h\lambda x(k + 1) = x(k)/(1 - h\lambda)$. Thus the region of stability for the backward Euler algorithm is $|1 - h\lambda| > 1$. Hence the backward algorithm provides a larger region of stability.

Approximation versus true solution and instability To maximize computational efficiency in solving a differential equation, we may consider it desirable to minimize the number of time steps by increasing the step size. However, choosing a step size is not easy. We will first demonstrate how the step size used in integration can create chaotic instability in the solution of an otherwise simpler problem. Consider the differential equation (initial value problem) $x'(t) = rx(t)(1 - x(t))$; $x(0) = x_0$ (Krishnamurthy and Sen 2001 (Chap. 9)). This is formally a nonlinear differential equation. However, it can be related to a linear problem and solved analytically by substituting $y(t) = 1/x(t)$. Then we get $y'(t) = -ry(t) + r$ and $y(0) = y_0$. This linear differential equation has the solution $y(t) = 1 - (1 - y_0)e^{-rt}$. Hence we obtain, for $0 < y_0 < 1$, the inequality $0 < y(t) < 1$ for all time $t > 0$ and for the parameter $r > 0$. Also the solution approaches the constant 1 as y(t) tends to unity when t tends to infinity. Similarly, if $y_0 > 1$ then $y(t) > 1$ for all $t > 0$ and again y(t) tends to unity as t tends to infinity.

We can now use the change of variable $x(t) = 1/y(t)$ and the solution of the differential equation is

$$x(t) = 1/[1 - \{1 - (1/x_0)\}e^{-rt}\}] = x_0e^{rt}/(x_0e^{rt} - x_0 + 1).$$

Thus the nature of the result obtained is independent of the parameter r although it can affect the rate of convergence. The solution, for any given initial value $x_0 > 1$, decreases monotonically to a value $x(t) = 1$. For $x_0 < 1$, it increases monotonically to a value $x(t) = 1$.

However, we will now consider approximation and the numerical solution of the above equation with a time step size $\Delta t = h$

$$x'(t) = (x(t + \Delta t) - x(t))/\Delta t = rx(t)(1 - x(t)) \text{ and } x(0) = x_0.$$

We will first show that this equation is identical to the logistic equation for $\Delta t = h = 1$. For this purpose we consider the logistic equation of the form $z(n+1) = az(n)(1 - z(n))$. We will modify it to an equivalent difference equation of the form $x(n+1) = x(n) + rx(n)(1 - x(n))$ by replacing $z(n) = r\,x(n)/(r + 1)$ and $a = r + 1$, i.e., $x(n + 1) - x(n) = rx(n)(1 - x(n))$.

Assuming that this is a difference equation with step Δt, we can rewrite the equation as $[x(n+1) - x(n)]/\Delta t = rx(t)(1 - x(t))$. Note that if $\Delta t = h = 1$, we recover the original equation. Also we have $x(n+1) - x(n) = \Delta t \times rx(t)(1 - x(t))$.

We showed in cobweb analysis, that the value of the parameter a in the logistic equation is very critical in deciding the type of the orbit and the convergence properties, when the parameter a changes its values to 2.9, 3.4, and 4. Now we need to replace the role of a by $\Delta t \times r$. *Hence the step size now becomes important*. We saw that for $a = 4$ we have chaotic orbits and convergence takes place for $a = 3$. Since we have $a = r + 1$, we have then chaotic orbits for $\Delta t = 1$ and $r = 3$ and convergence takes place for $r < 2$. In the case of difference equation this means we need to restrict the step size so that

$$\Delta t \times r < 2 \text{ or } h = \Delta t < 2/r.$$

This is a *stability condition* arrived at using chaos theory. *The above example illustrates that the translation of a differential equation problem to its numerical approximation is very delicate and can be extremely sensitive to small errors . In this approximation process, we may inadvertently change the problem significantly, leading to unexpected outcomes that may require an entirely new theoretical approach for the interpretation of the results obtained.*

Remark One of the crucial differences between the discrete and the continuous version of the logistic equation we used is that in the continuous case chaos is impossible in dimensions 1 and 2 (Peitgen et al. 1991).

Chaotic or strange attractors While dealing with iterative dynamics for solving equations we saw that the attractors are generally points, cycles, or smooth surfaces having integral geometric dimensions. However, there are other attractors that are poorly understood in terms of their geometry and dynamics. These attractors have fractional geometric dimensions (fractals) and hence called chaotic or strange attractors. In this section our aim is to introduce these strange attractors in the context of numerical solution of differential equations.

Rossler system A system of simultaneous differential equations to demonstrate the presence of chaos is the Rossler system. This system is given by

$$dx/dt = x' = -(y + z),\ dy/dt = y' = x + ay,\ dz/dt = z' = b + xz - cz.$$

The behaviour of this system has been shown to be chaotic by showing how the stretching and folding or baker transformation is embedded in the system. In fact it is one of the simplest systems that exhibit folding action in three dimensional phase space and it is a spiral attractor.

Taking $a = b = 0$ and varying $c > 0$, we can plot the behaviour in three axes X, Y, and Z , as c is increased. Renaming the variables t, x, y, z as x, y_1, y_2, y_3, respectively (for generalization) and setting $a = b = 0$, we have the foregoing simultaneous system as

$$y'_1 = -(y_2 + y_3),\ y'_2 = y_1,\ y'_3 = y_1 y_3 - c y_3$$

with initial condition $y_1 = 4$, $y_2 = 0$, $y_3 = 0$ at $x = 0$. By choosing $c = 2$ and $x = 0(0.5)2$ creating the Matlab M-file

```
function yd=f(x,y);
c=2; yd=[-(y(2)+y(3)); y(1); y(1)*y(3)-c*y(3)];
```

and then using **ode23** as

>>xspan=0:.5:10; y0=[4; 0; 0]; [x, y]=ode23('yd', xspan, y0); soln=[x,y]; plot(x,y)

we get the graph (Fig. 19)

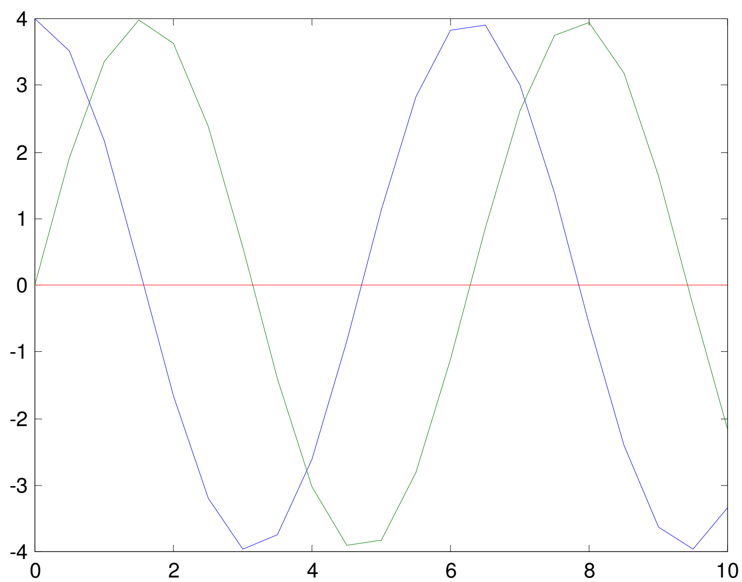

Fig. 19 Graph of the solution of odes $y'_1 = -(y_2 + y_3)$, $y'_2 = y_1$, $y'_3 = y_1 y_3 - c y_3$ with $c = 2$ and initial condition $y_1 = 4$, $y_2 = 0$, $y_3 = 0$ at $x = 0$ with 20 steps $x = 0(0.5)2$ obtained using the M-file and the commands containing containing **ode23**

Using the same M-file

```
function yd=f(x,y); c=2; yd=[-(y(2)+y(3)); y(1); y(1)*y(3)-c*y(3)];
```

and then using the Matlab commands (for 3-dimensional plot)

>>xspan = 0: 1: 200; y0 = [4; 0; 0]; [x, y] = ode23('yd', xspan, y0); soln = [x, y];
ya = y(:, 1); yb= y(:, 2); plot3(x, ya, yb)

we obtain Fig. 20.

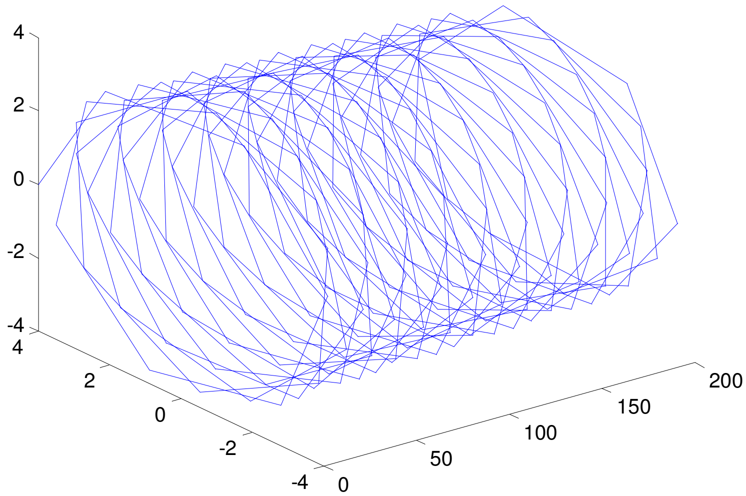

Fig. 20 Graph of the solution of odes $y'_1 = -(y_2 + y_3)$, $y'_2 = y_1$, $y'_3 = y_1y_3 - cy_3$ with c = 2 and initial condition $y_1 = 4$, $y_2 = 0$, $y_3 = 0$ at x = 0 with 200 steps x = 0(1)200 obtained using the M-file and the commands containing containing **ode23**

This system undergoes a series of period doublings between c = 2.5 and c= 5.0. If we now replace c = 2 by c = 3 in the foregoing M-file, **ode23** by **ode45** in the foregoing Matlab commands and take the steps x = 0(1)200, we obtain Fig. 21. The required M-file is

```
function yd=f(x,y); c=3; yd=[-(y(2)+y(3)); y(1); y(1)*y(3)-c*y(3)];
```

and the Matlab commands using **ode45** are

>>xspan=0:1:200; y0=[4; 0; 0]; [x, y]=ode45('yd', xspan, y0); soln=[x,y]; plot(x,y)

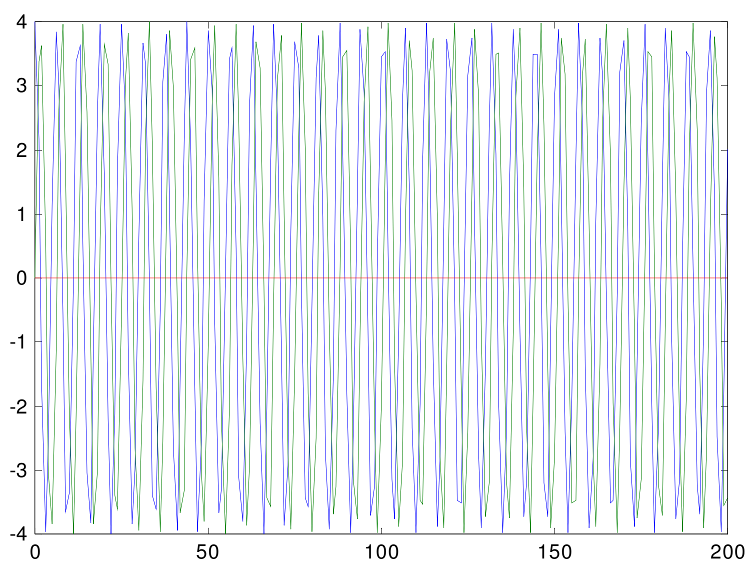

Fig. 21 Graph of the solution of odes $y'_1 = -(y_2 + y_3)$, $y'_2 = y_1$, $y'_3 = y_1y_3 - cy_3$ with c = 3 and initial condition $y_1 = 4$, $y_2 = 0$, $y_3 = 0$ at x = 0 with 200 steps x = 0(1)200 obtained using the M-file with c = 3 and the commands containing **ode45** and **plot**

Using the same M-file, viz.,

```
function yd=f(x,y); c=3; yd=[-(y(2)+y(3)); y(1); y(1)*y(3)-c*y(3)];
```

and the Matlab commands (that includes 3-dimensional plot command **plot3**)

```
>>xs = 0: 1: 200; y0 = [4; 0; 0];[x, y] = ode45('yd', xs, y0); soln = [x, y];
>>ya = y(:, 1); yb = y(:, 2); plot3(x, ya, yb)
```

we obtain Fig. 22.

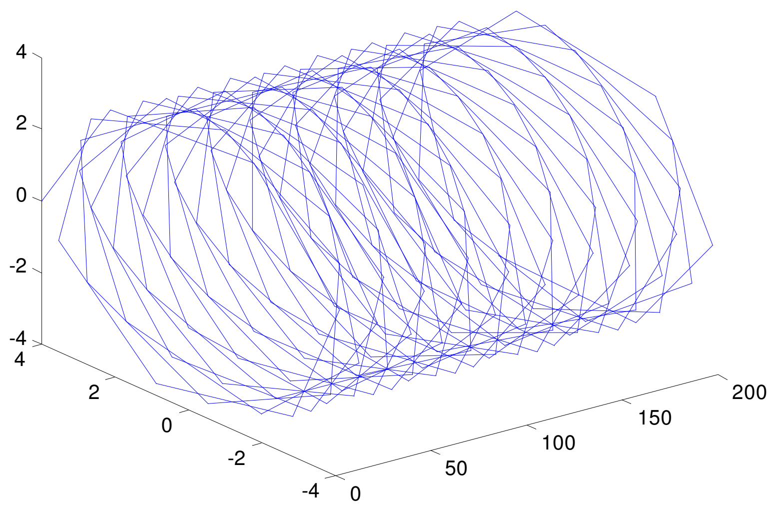

Fig. 22 Graph of the solution of odes $y'_1 = -(y_2 + y_3)$, $y'_2 = y_1$, $y'_3 = y_1y_3 - cy_3$ with c = 3 and initial condition $y_1 = 4$, $y_2 = 0$, $y_3 = 0$ at x = 0 with 200 steps x = 0(1)200 obtained using the M-file with **c = 3** and the commands containing **ode45** and **plot3**

With an initial condition $x(0) = 4$, $y(0) = z(0) = 0$ and varying t from 0 to 200 we see that beyond c = 5.0 produces chaotic attractor. The trajectory spends most of its time in the X-Y plane spiralling out of the origin until some critical value and then jumps away from the plane, moving in the z direction until some maximum value is obtained and then dropping back into the $x - y$ plane to undertake a new spiralling out motion and so on. This attractor has a fractal dimension and hence called a “strange attractor”.

To find the equilibrium point of the Rossler system we set $x' = y' = z' = 0$. This provides two fixed points for b =2, c = 4 and $0 < a < 2$. For a very small z the motion near the origin is an unstable spiral.

Lorenz system This model was proposed by Lorenz in 1963 to model the weather pattern. Using this model as the basis Lorenz proved the impossibility of weather prediction beyond 2 weeks by showing the presence of chaotic attractor (strange attractor) in the numerical study of a set of differential equations. Lorenz model is specified by the three equations

$dx/dt = x' = -ax + ay$, $dy/dt = y' = -xz + rx - y$, $dz/dt = z' = xy - bz$.

Note that the second and third equations are nonlinear unlike in Rossler system where only one of the equations is nonlinear.

Continuous case The Lorenz system has three possible steady states obtained by setting $x' = y' = z' = 0$. Using the first equation $x = y$, we have from the third equation $z = x^2/b$; using this z in the second equation, we get: $x^3 - rbx + bx = 0$ or $x(x^2 + b(1 - r)) = 0$. Thus the steady states are $S(1) = (x, y, z) = (0, 0, 0)$. The two other equilibrium states S(2) and S(3) exist for $r > 1$ and are given by $S(2) = (\sqrt{[b(r - 1)]}, \sqrt{[b(r - 1)]}, (r - 1))$; $S(3) = (-\sqrt{[b(r - 1)]}, -\sqrt{[b(r - 1)]}, (r - 1))$. The stability of the steady state is given by the eigenvalues of the following Jacobian matrix of the partial derivatives

$$J = \begin{bmatrix} -a & a & 0 \\ r - z & -1 & -x \\ y & x & -b \end{bmatrix}$$

evaluated at the state $S(1) = (0, 0, 0)$. This gives the evaluated Jacobian matrix

$$J_{S(1)} = \begin{bmatrix} -a & a & 0 \\ r & -1 & 0 \\ 0 & 0 & -b \end{bmatrix}$$

The characteristic equation of this matrix is $(b + \lambda)[(a + \lambda)(1 + \lambda) - ar] = 0$. Thus the eigenvalues are $\lambda(1) = -b$; $\lambda(2) = 0.5[-(a + 1) - \sqrt{[(a - 1)^2 + 4ar]}]$, $\lambda(3) = 0.5[-(a + 1) + \sqrt{[(a - 1)^2 + 4ar]}]$. Let us now fix the values of a and b to be $a = 10$; $b = 8/3$ and examine the behaviour of the Lorenz system as r is varied in the range $0 < r < 1$. The eigenvalues are negative and S(1) is the only attractor of the system. This is because $(a + 1) \geq \sqrt{[(a - 1)^2 + 4ar]}$ as long as $0 < r < 1$.

As r passes through 1, one of the eigenvalues of S(1) becomes positive with the other two remaining negative. This indicates that S(1) has a 2-dimensional stable manifold and a 1-dimensional unstable manifold. Thus S(1) becomes a saddle point.

When $r > 1$, the two new fixed points S(2) and S(3) with the values specified above, are born. They are stable at birth. Let us denote $d = \sqrt{[b(r - 1)]}$. Then $S(2) = (d, d, r - 1)$; $S(3) = (-d, -d, r - 1)$. We can now evaluate the eigenvlues of the Jacobian at S(2) and S(3). The Jacobian at S(2) is

$$J_{S(2)} = \begin{bmatrix} -a & a & 0 \\ 1 & -1 & -d \\ d & d & -b \end{bmatrix}$$

Its characteristic equation is $\lambda^3 + \lambda^2(a + b + 1) + \lambda(ab + br) + 2ab(r - 1) = 0$. To study this system we need to fix the values of a, b, and r. Setting a = 10, b = 8/3 we can vary r and study this system.

As r becomes greater than 1, at a particular value the two negative eigenvalues of S(2) and S(3) coalesce and become complex conjugate eigenvalues with negative real parts. In this regime the orbits approach S(2) and S(3) by spiraling around them. As r is increased further the spirals increase in size until some critical value r(0) = 13.96. As r increases further to r= 24.06 the states S(2) and S(3) become unstable as the real parts of their complex conjugate eigenvalues pass from negative values to positive values (a bifurcation) and chaotic instability sets in.
In summary we find that, for

(i) $0 < r < 1$, there is only one fixed point at the origin,
(ii) $1 < r < 1.346$, two new stable nodes are born, and the origin becomes a saddle point with a one dimensional unstable manifold,
(iii) $1.346 < r < 13.926$, at the lower value, the stable nodes become stable spirals,
(iv) $13.926 < r < 24.74$, instability sets in and the two fixed points become intertwined,
(v) $24.74 < r$, all three fixed points are unstable, chaos sets in, and the origin is a saddle point while the other two points are spiral equilibrium points.

The orbit in case (v) has the appearance of the two wings of a butterfly; for small changes in the parameters, the orbit can switch from one wing to the other. This situation is known as the "butterfly effect". It is best to view the above situation and the dynamics of the trajectories in graphics supported by Matlab.

Discrete case In order to solve the Lorenz system in discrete form it is not necessary to recompute the eigenvalues of the discretized version, since the information that the real part of the eigenvalues $Re[\lambda(i)] < 0$ of the continuous system implies that eigenvalues of the discretized version lies within the unit circle as long as the time step for discretization is chosen adequately small.

5. Conclusions

Measuring sensitivity and stability of fixed points In numerical computation stability is of supreme importance and so suitable measures are needed for this purpose. As we have been progressing through our study we realise that the eigenvalues of the Jacobian matrix at the fixed points play a crucial role in determining the stability of the fixed points. Lyapunov exponents and Lyapunov numbers are respectively related to the eigenvalues and characteristic multipliers. These are extremely useful in determining the sensitive dependence on initial conditions, stability of the fixed points, periodic orbits, and the chaotic behaviour.

The Lyapunov exponent quantifies the average growth of an infinitesimally small error in the initial point. If we assume that the initial error is E(0) we will be interested to find out how the error E(n) grows at the nth functional application. Assuming that the rate of growth of error is of the form $E(n)/E(0) = e^{bn}$ or $\log E(n)/E(0) = bn$ or $b = (1/n) \log (E(n)/E(0))$. (Another practical way to measure the growth of the error is to assume the rate to be 2^{bn} so that we can directly compute the error growth in bits). If a function is iterated n times and the initial error is E(0) and after n steps the error is E(n), then a reasonable way to measure amplification or contraction of errors is to use the function $L= (1/n) \log |E(n)|/|E(0)|$ which can be expressed approximately by $(1/n) \Sigma \log |E(k)/E(k-1)|$; the summation Σ is taken over k = 1 to n.

This measure is called the Lyapunov exponent and is related to the eigenvalues of the matrix of transformation. The rate of growth is related to the derivative of f(x) at each point for a one variable map and the eigenvalues of the Jacobian for multivariable maps. This is explained below.

Consider the slope between two points x(0) and x(1) or, equivalently, the first derivative df/dx at x = x(0), i.e., [f(x(1) − f x(0)]/[x(1) − x(0)] = s(1)/s(0) = f′(x(0)). In order to work out the global or total rate of change of the functional $f^n(x)$ when f is applied to itself n times, chain defferentiation rule can be used. We can show using the rule that when f is applied n times and the trajectory passes through x(1), . . ., x(n − 1) we get

$$d/dx\, f^n(x) = f'(x(n-1)) \times f'(x(n-2)) \times \ldots \times f'(x(0)).$$

As an illustration of the rule, consider $f(x) = 2x^3 - 1$. Then $f^2(x) = f(f(x)) = f(2x^3 - 1) = 2(2x^3 - 1)^3 - 1$. The derivative $df^2(x)/dx = 6(2x^3 - 1)^2 \times 6x^2$. Allowing $x(1) = 2x(0)^3 - 1$ and x(0) = x, we obtain $df^2(x)/dx|_{x=x(0)} = 36x(1)^2x(0)^2$. We also have $f'(x(1)) = 6x(1)^2$ and $f'(x(0)) = 6x(0)^2$. Hence $f'(x(1))f'(x(0)) = 36x(1)^2x(0)^2$.

If we assume that the neighbouring trajectories are separated in an exponential manner, we can write the nth step separated by the ratio

$$s(n)/s(0) = \exp bn = d/dx\, f^n(x) = f'(x(n-1)) \times f'(x(n-2)) \times \ldots \times f'(x(0)).$$

The nth root of this ratio, i.e., the *n*th root (also called the geometric mean) of this product is the *Lyapunov number*. The coefficient of change b is given by (1/n) times the logarithm of the product, i.e., $b = (1/n)\Sigma \log f'(x(i))$, where the summation extends from 0 to (n − 1) and n is assumed to be very large, tending to infinity. The natural logarithm (base e) of the Lyapunov number is the *Lyapunov exponent* and measures the average of the log of the absolute values of the n first order derivatives of the local slopes, as measured over the entire attractor or fixed point.

Lyapunov exponents, numbers, and eigenvalues For continuous systems, let *J*(1),*J*(2), . . .,*J*(n) be the eigenvalues of the Jacobian matrix or, simply, Jacobian *J*F(x). Then Lyapunov exponents are defined as the real parts of the eigenvalues at the equilibrium point. They indicate the rate of contraction (*J*(i) < 0) or expansion (*J*(i) > 0) near the equilibrium point when t tends to infinity (in the limit).

For discrete systems we use the Lyapunov numbers. Here the equilibrium point is called a fixed point of the map $F^k(x) = x^*$. Since the Jacobian of $F^k(x^*) = [\,JF(x^*)^k]$, the eigenvalues of this matrix are denoted by $m^k(1), m^k(2), \ldots, m^k(n)$. The Lyapunov numbers of the fixed point are $m(i) = \lim k \rightarrow \infty\, |m(i)^k|^{1/k} = |m(i)|$.

It can be shown that this corresponds to the exponential of the product of the eigenvalues of the Jacobian multiplied by the time period of sampling h. That is $|m(i)| = [\exp \lambda(i)h]$ and hence the Lyapunov exponent $\lambda(i) = (1/h)\log |m(i, h)|$. Usually this value is summed over an entire interval T= nh around the fixed point, and the average is taken, i.e., $\lambda(i) = (1/n)\Sigma \log |m(i, h)|$. Thus λ(i) gives the rate of contraction and expansion while m(i) is the amount of contraction or expansion in a time period h. Since

$$|m(i)| < 1, \text{ if and only if } Re[\lambda(i)] < 0 \text{ and } |m(i)| > 1 \text{ if and only if } Re[\lambda(i)] > 0,$$

they provide the same stability information. Thus if $\text{Re}[\lambda(i)] < 0$ for a continuous system, it implies that $|m(i)|$ lies within the unit circle for the corresponding discretized version, as long as the time step for discretization is chosen adequately.

Linear system Consider the solution of the linear system Ax = b, in which we decompose A into the form M – N and where M is invertible. Then we can write $x = M^{-1}(Nx + b)$ and the iterative process takes the form $x(i + 1) = M^{-1}(N(x(i) + b))$.

Usually one cannot compute M unless it is easily invertible as in the diagonal case of the Gauss-Seidel or the Jacobi iteration method (Krishnamurthy and Sen 2001). In the case of the Gauss-Seidel method we choose M = D – E and N = –F, where D is the diagonal part of A, –E is the lower sub-triangular part and –F is the upper sub-triangular part. The Jacobi method chooses M = D and N = E + F where –E and –F are the lower sub-triangular and the upper sub-triangular parts of A. The successive over relaxation method chooses M = (1/w)D – E; and N = ((1/w) – 1)D + F where –E and –F are the lower sub-triangular and upper sub-triangular parts of A and w a relaxation parameter.

Since we want the iterative sequence converge, we need $\lim i \to \propto (M^{-1}N)^i x(0) = 0$. This means that we need the spectral radius $r(M^{-1}N) < 1$ in magnitude. Note that $\log r(M^{-1}N)$ is negative and it essentially estimates the Lyapunov exponent and tells us how the trajectory converges. For example, given two converging methods with spectral radii r^* and r^{**}, we can say one converges faster than the other if $r^* < r^{**}$. If in fact $r^{**} = 2r^*$, then the first method with spectral radius r^* converges twice as fast. This simple example illustrates the utility of Lyapunov exponents for measuring the rate of expansion or contraction of trajectories.

Lyapunov dimension Let $\lambda(1) \geq \lambda(2) \geq \ldots \geq \lambda(n)$ be the Lyapunov exponents of the attractor of a continuous dynamical system. Let j be the largest integer such that $\lambda(1) + \lambda(2) + \ldots + \lambda(j) \geq 0$. Then the Lyapunov dimension is defined by $D(L) = j + \{\lambda(1) + \lambda(2) + \ldots + \lambda(j)\}/|\lambda(j + 1)|$. If no such j exists, D(L) = 0 (For a proof, see Parker and Chua 1989). Thus for an attractor all whose Lyapunov exponents are given by $0 > \lambda(1) \geq \lambda(2) \geq \ldots \geq \lambda(n)$, we get the Lyapunov dimension D(L) = 0. For a periodic state (one or more points), where $\lambda(1) = 0$ and $0 > \lambda(2) \geq \ldots \geq \lambda(n)$, the Lyapunov dimension D(L) =1. For a two periodic case (one or more closed curves), where $\lambda(1) = \lambda(2) = 0$ and $0 > \lambda(3) \geq \ldots \geq \lambda(n)$, the Lyapunov dimension D(L) = 2.

For the nonchaotic attractor, they are points, cycles or smooth surfaces with a regular geometry. However, when the dynamics is chaotic, the attractor is called strange or chaotic and the trajectories around the attractor are chaotic leading to a complex geometrical object having a nonintegral dimension. The Lyapunov dimension is a non-integer representing a fractal dimension. For example, if $\lambda(1) > 0 > \lambda(2)$, then $2 < D(L) = 2 + \lambda(1)/|\lambda(2)| < 3$.

The main use of the dimension is to quantify the complexity of an attractor. It gives a lower bound on the number of state variables required to describe the dynamics of the system or the number of degrees of freedom of the attractor.

For a Lorenz attractor with a =16, r = 45.92, and b = 4 the three eigenvalues are 2.16, 0.00, and –32.4. This gives the Lyapunov dimension 2 + (2.16/32.4) = 2. 066.

Chaos and Lyapunov exponent Chaos in dynamics implies sensitivity to initial conditions. If one imagines a set of initial conditions within a sphere of radius in the phase space ,then for chaotic motion the trajectories originating in the sphere will map the sphere into an ellipsoid whose major axis grows as d= r exp λ t where λ is called the Lyapunov exponent. $\lambda > 0$ for chaotic motion and $\lambda < 0$ for regular motion. Thus the sign of λ is a criterion for chaos and can be used to measure the rate of growth or contraction of a function that is sensitive to errors in initial conditions in a manner analogous to measuring the ill-conditioning of a matrix (Chapter 5, Krishnamurthy and Sen 2001).

Examples (i) Consider $F(x) = 5/2\, x(1 - x)$. Let $x(0) = 0$; the fixed points of F(x) are 0 and 3/5. Here $F'(x) = 5/2$ at $x = 0$; $F^n(0) = 0$ for all n. We have the Lyapunov number $N(0) = \lim n \to \propto\ {}^n\sqrt{(5/2)^n} = 5/2$. The dominant Lyapunov exponent is log N(x(0)) = log 5 – log 2.

Let $x(0) = 3/5$; then $F'(x(0)) = 5/2 - 5x = -1/2$ at $x(0) = 3/5$. The Lyapunov number at $x(0)= 3/5$ is $N(x(0))$ = Lim n tends to infinity ${}^n\sqrt{(1/2)^n} = (1/2)$; hence the Lyapunov exponent is –log 2.

(ii) Consider the parameterized tent map

$$T(x) = 2rx(n + 1) = 2x(n) \text{ for } 0 \le x(n) \le 1/2$$
$$= 2r(1 - x(n)) \text{ for } ½ < x(n) \le 1.$$

Here $|T'(x)| = 2r$; the Lyapunov exponent equals log 2r. Thus when $2r > 1$, the Lyapunov exponent is greater than 0 and the motion is chaotic. When $2r < 1$, the Lyapunov exponent is less than 0 and the motion is regular and all the points are attracted to $x = 0$.

Dynamical systems and computability theory A dynamical system is a system that evolves with time. These systems are classified as linear and nonlinear depending upon respectively,whether the output is a linear or nonlinear function of the input . A linear system analysis (as with linear equations) is straightforward, while the analysis of a general nonlinear system is complicated because one cannot simply decompose the system into Input + Transfer function = Output as in a linear system. This because nonlinear systems can be dissipative, i.e., they lose some energy to irreversible processes such as friction. Accordingly, nonlinear systems can have attractors of four kinds, viz., fixed or equilibrium points, periodic orbits, quasi periodic attractors, and chaotic or strange attractors.

Small parameter changes in nonlinear sytems can lead to abrupt change in behaviour, e.g bifurcation. Finding attractors and the basins are important issues in nonlinear dynamical systems. Lyapunov exponent serves as an important invariant to quantify the behaviour of an attractor. A system with negative Lyapunov exponents imply stability and a system with the positive Lyapunov exponents implies instability.

Thus in a formal sense, all our earlier discussion on the concept of points such as attractors, periodic points with respect to nonlinear system of equations can be translated in the context of the analysis of nonlinear dynamical systems.

We also add that most classical dynamical systems are associated with regular periodic motion and the associated differential equations are solvable. Such dynamical systems are called Integrable systems. When a system is nonlinear it turns out to be nonintegrable and exhibits various degrees of irregular dynamics:

(i)Ergodicity Here the set of points in phase space behave in such a way that the time-average along a trajectory equals the ensemble average over the phase space. Although, the trajectories wander around the whole phase space , two neighbouring initial points can remain fully correlated over the entire time of evolution.

(ii) Mixing The initial points in the phase space can spread out to cover it in time but at a rate weaker than the exponential (e.g. inverse power of time).

(iii) K-flow or Chaos The trajectories cover the phase space in such a way that the initially neighbouring points separate exponentially and the correlation between two initially neighbouring points decays with time exponentially. It is with this class of irregular motion we define classical chaos.

Each of the above properties imply the all the properties above, e.g., within a chaotic region the trajectories are ergodic on the attractor and wander around the desired periodic orbit. As explained in previous sections , classical motion is chaotic if the flow of the trajectories in a given region of phase space has positive Lyapunov exponents that measure the rate of exponential separation between two neighbouring points in the phase space. Chaos indicates hypersensitivity on the initial conditions. Also the system becomes inseparable (metric transitivity) and the periodic points are dense. That is the whole dynamical system is simply not a sum of parts and it functions as a whole leading to what is known as "Emergence". Also chaotic dynamic systems are governed by strange attractors having fractal dimension!

The edge of chaos and computability We saw in this paper that solving differential equations (equivalent to "Integrability") and attractor finding (equivalent to "zero or root- finding") are fundamental issues in dynamical systems. These two operations are analogous to the primitive recursion operation and zero-testing operation used in computability theory (Krishnamurthy 1983).

Also we observed that a fixed point (equilibrium) or an attractor corresponds to finding a stable condition in a dynamical system. Similarly, finding fixed point in a recursive program implies termination and computability. When a problem remains in the border of computability and noncomputability, we do not know whether a fixed point exists and we are at the edge of computability analogous to entering a route to chaos in dynamics. Thus chaos in dynamical systems and noncomputability can be considered as parallels in their respective domains. Undecidable partially recursive schemes of number theory also exhibit chaos-like behaviour, such as lack of predictability and decidability! In fact, it is suspected that deterministic chaos corresponds to Godel's undecidability. An example of a recursive sequence (called Ulam sequence) that illustrates this situation is provided below.

Example Starting from any natural number N, the elements of sequence are generated by the recursive sequence

while $N(i) \neq 1$, *if* $N(i)$ *is even then* $N(i+1)=N(i)/2$; *else* $N(i+1)=3N(i)+1$.

This conjecture is due to Ulam; it states that this sequence will terminate after a finite, possibly arbitrarily large number steps k, resulting in $N(k) = 1$. Ulam's conjecture is as yet unproved although verified for a very large N. Ulam's conjecture corresponds to finding the fixed point of the following recursive program

P: *F(N):= if N=1 then 0 else if even(N)then F(N/2) else F(3N+1)*

may or may not have a fixed point for all N. The whole program can again be rewritten as

$$N(i + 1) = 1/2\ N(i)\ (5N(i) + 3) - (5N(i) + 2)\ [.5\ N(i)],$$

where [] denote integral part of the quantity not exceeding it. This problem of deciding N(k) = 1 for a large N(0) is computationally intractable since we need a global information of the first term and this takes an exponential amount of time! Thus this problem is of great interest since it lies in the border of intractability and undecidability.

It is also interesting to note that this problem can be rearranged so that it contains as a sub-problem the Bernoulli shift which leads to chaotic dynamics, namely

$$F(X(i)) = X(i + 1) = 2X(i) \bmod 1,$$

where a mod 1 = a − Integer (x) = fractional part. For this function $F'(X) = 2$ is constant and on the average the distance between the nearby points grows as $d(n) - d(0)2^n$. After m iterations we have lost the knowledge of the initial state. The Lyapunov exponent is $1/n \log 2^n = 1$ and the motion is chaotic.

Final remarks In this paper we explained a very simple class of dynamical systems related to the numerical iterated function systems which can exhibit chaos. We also mentioned that an analogous situation arises in recursive algorithms leading to intractability and undecidability. In other words, both dynamical systems as well as computing systems can lead to very complex and unpredictable outcomes in spite of the fact that they are deterministic. This aspect forms the basis for the study of a new kind science, called " *complex systems*" (Wolfram 2002) that explores the interrelationship among many different sciences so as to understand the universality principles that govern them.

Iterated function systems also provide a convenient framework for constructing and understanding fractal objects as well as neural network model. From a numerical analysis point of view the neural network model is an iterative dynamical model. Interactive software based computing with sophisticated graphic facilities, simulation and emulation play a major role in these exploratory areas, which are largely experimental in nature. Our introduction to these challenging new areas is very rudimentary and the readers are encouraged to read further, the important references cited here.

Acknowledgement The authors wish to thank Professor E.V. Krishnamurthy, Australian National University, Canberra not only for initiating one of the authors (S.K. Sen) in chaotic dynamical systems but also providing them enormous amount of information in this area. Also, this article has extensively used the unpublished material authored jointly by him (E. V. Krishnamurthy) and S.K. Sen (one of the authors of the present article).

References

Baker, G.L.; Gollub, J.P. (1996): Chaotic Dynamics: An Introduction, Cambridge University Press, Cambridge.

Davies, B. (1999): Exploring Chaos, Perseus Books, Reading, Massachusetts.
Devaney, R.L. (1989): Chaotic Dynamical Systems, Addison Wesley, Reading, Massachusetts.
Enns, R.H.; McGuire, G.C. (1997): Nonlinear Physics, Birkhuser, Berlin.
Flake, G.W. (2001): The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Application, MIT Press, Cambridge, Massachusetts.
Holmgren R.A. (1994): A First Course in Discrete Dynamical Systems, Springer, New York.
Kaye, B. (1993): Chaos and Complexity, VCH, New York.
Lam, L. (1999): Nonlinear Physics for Beginners,World Scientific, Singapore.
Lorenz, E.N. (1963): Deterministic non-periodic flow, *J. Atmos. Sci.***20**, 130-41.
Martelli, M. (1992): Discrete Dynamical Systems and Chaos, Longmans, Harlow, Essex, U.K.
Ott, E.; Sauer, T.; Yorke, J. (1994): Coping with Chaos, Wiley, New York.
Ott, E. (2003): Chaos in Dynamical Systems, Cambridge University Press,Cambridge.
Parker,. T.S.; Chua, L.O. (1989): Practical Numerical algorithms for chaotic systems, Springer Verlag, New York.
Peitgen, H. (1989): Newton's method and Dynamical systems, Kluwer, London.
Peitgen, H.; Jurgens, H.; Saupe, D. (1991): Fractals for the Class Room, Parts 1&2, Springer Verlag, New York.
Robert, F. (1986): Discrete Iterations, Springer Verlag, New York.
Wiliams, G. P. (1997): Chaos Theory Tamed, Joseph Henry Press, Washington, D.C.
Wolfram, S. (2002): A New Kind of Science, Wolfram Media Inc., Champaign, Illinois.