Приднестровский государственный университет им. Т. Г. Шевченко

Физико-математический факультет

# **Контрольная работа**

# по дисциплине: **Основы теории динамических систем**

**«Бифуркационное дерево»**

Выполнила: Студентка 503 группы ФМФ Слободянюк А.А.

Проверил: доцент Соковнич С.М.

Тирасполь 2010

**Содержание**

Введение

1. Бифуркационное дерево

2. Постановка задачи

Литература

Приложение

**Введение**

К хаосу системы могут переходить разными путями. Среди последних выделяют бифуркации, которые изучает теория бифуркаций. Бифуркация (от лат. bifurcus - раздвоенный) представляет собой процесс качественного перехода от состояния равновесия к хаосу через последовательное очень малое изменение периодических точек.

Мы знаем из определения, бифуркации возникают при переходе системы от состояния видимой стабильности и равновесия к хаосу. Примерами таких переходов являются дым, вода и многие другие самые обычные природные явления. Так, поднимающийся вверх дым сначала выглядит как упорядоченный столб. Однако через некоторое время он начинает претерпевать изменения, которые сначала кажутся упорядоченными, однако затем становятся хаотически непредсказуемыми. Фактически первый переход от стабильности к некоторой форме видимой упорядоченности, но уже изменчивости, происходит в первой точке бифуркации. Далее количество бифуркаций увеличивается, достигая огромных величин. С каждой бифуркацией функция турбулентности дыма приближается к хаосу. С помощью теории бифуркаций можно предсказать характер движения, возникающего при переходе системы в качественно иное состояние, а также область существования системы и оценить ее устойчивость.

**1. Бифуркационное дерево**

Характерной иллюстрацией возможных бифуркаций в системе служит бифуркационное дерево, представляющее собой зависимость возможных дискретных значений динамической переменной на аттракторе от параметра. Типичный пример такого дерева, построенного с помощью компьютера, показан на рисунке. Приведенный пример относится к одной из эталонных моделей нелинейной динамики – логистическому отображению. Наше аналитическое рассмотрение позволяет нарисовать начальный участок дерева (рис. 1)

Рис. 1. Зависимость установившегося значения переменной от параметра при бифуркации удвоения периода.

На нем изображены устойчивая неподвижная точка и рождающийся 2-цикл. В последнем случае переменная последовательно посещает две ветви дерева. Такую ситуацию расщепления дерева называют бифуркацией удвоения периода. Полное дерево (для всех значений параметра) можно построить с помощью компьютера. Для этого надо задать некоторое начальное значение переменной и параметра. Затем выполнить несколько сот итераций отображения, чтобы исключить переходные процессы и реализовать установившийся режим, и вывести некоторое количество точек на экран дисплея. Затем процедуру повторить для слегка измененного количества параметра (рекомендуем в качестве нового начального значения переменной использовать полученное на предыдущем шаге процедуры) и продолжать повторять до тех пор, пока весь интересующий диапазон значений управляющего параметра не будет пройден. В результате получится картинка, показанная на рис. 2.

На бифуркационном дереве хорошо видны моменты удвоений периода, в которые дерево расщепляется на две ветви, хаотический режим и различные «окна» периодических режимов в хаосе.

Тщательное рассмотрение бифуркационного дерева приводит к интересному выводу: внутри окон периодичности также наблюдаются удвоения периода. Например, при λ=1,75 возникает устойчивый 3-цикл. Однако с ростом λ он исчезает на дереве (на самом деле становится неустойчивым), и от него отделяется устойчивый 6-цикл, затем 12-цикл и т.д. Каскад удвоений 3-цикла завершается переходом к хаосу в своей, новой критической точке. То же самое относится и к циклам других периодов.

Рис. 2. Бифуркационное дерево логистического отображения

**2. Постановка задачи**

Создать программу, которая при щелчке мышью на бифуркационном дереве в отдельном окне строит итерационную диаграмму.

**Литература**

## 1. И. Ануфриев, А. Смирнов, Е. Смирнова «MATLAB 7» Cанкт-Петербург 2005 г

## 2. А.П. Кузнецов, А.В. Савин, Л.В. Тюрюкина «ВВЕДЕНИЕ В ФИЗИКУ НЕЛИНЕЙНЫХ ОТОБРАЖЕНИЙ» Саратов 2010 г

## 3. Ж.Йосс, Д.Джозеф « ЭЛЕМЕНТАРНАЯ ТЕОРИЯ УСТОЙЧИВОСТИ И БИФУРКАЦИЙ »

1. бифуркация компьютерный итерационный периодический

**Приложение**

function varargout = mygui0(varargin)

% MYGUI0 M-file for mygui0.fig

% MYGUI0, by itself, creates a new MYGUI0 or raises the existing

% singleton\*.

%

% H = MYGUI0 returns the handle to a new MYGUI0 or the handle to

% the existing singleton\*.

%

% MYGUI0('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in MYGUI0.M with the given input arguments.

%

% MYGUI0('Property','Value',...) creates a new MYGUI0 or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before mygui0\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to mygui0\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help mygui0

% Last Modified by GUIDE v2.5 25-Dec-2010 20:41:17

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @mygui0\_OpeningFcn, ...

'gui\_OutputFcn', @mygui0\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before mygui0 is made visible. function mygui0\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to mygui0 (see VARARGIN)

% Choose default command line output for mygui0 handles.output = hObject;

% Update handles structure guidata(hObject, handles);

% UIWAIT makes mygui0 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line. function varargout = mygui0\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in btnPlot.

function btnPlot\_Callback(hObject, eventdata, handles)

% hObject handle to btnPlot (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

axes(handles.axes1)%задание текущего графика axes1

%

Построение бифуркационной диаграммы

n=10;

N=2^n;% Число итераций отображений

r0=0;% Начальная точка диаграммы

r1=2;% Конечная точка диаграммы

a(1)=0.1;% Нулевая итерация

Nb=100;%Число сохраненных итераций

Nr=1000;%Число шагов по паметру r

steppr=(r1-r0)/Nr;%Шаг по параметру r

for j=1:Nr %Цикл по параметру r

r(j)=r0+steppr\*j;

for i=2:(N-Nb)% Не сохраненные итерации

a(i)=logist(a(i-1),r(j));

end

b(j,1)=a(N-Nb);

for i=2:Nb%Не сохраненные итерации

b(j,i)=logist(b(j,i-1),r(j));

end

end

plot(r,b,'.b','MarkerSize',1);

% x = -2:0.2:2;

% y = exp(-x.^2);

% %plot(x, y)

% handles.Line=plot(x, y);

% guidata(gcbo, handles)%сохранение структуры handles для использования в других подфункциях

title('Axes 1');

xlabel('\lambda');

ylabel('X');

set(hObject, 'Enable', 'off')%кнопка очистить должна стать недоступной после вывода графика

set(handles.btnClear, 'Enable', 'on')% кнопка Очистить должна стать доступной

% --- Executes on button press in btnClear.

function btnClear\_Callback(hObject, eventdata, handles)

% hObject handle to btnClear (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

axes(handles.axes1);%задание текущего графика axes1

cla %Очистка осей

axes(handles.axes2);%задание текущего графика axes2

cla %Очистка осей

set(hObject, 'Enable', 'off')%кнопка очистить должна стать недоступной после очистки осей

set(handles.btnPlot, 'Enable', 'on')%кнопка Построить должна стать доступной

% --- Executes on mouse press over axes background.

function axes1\_ButtonDownFcn(hObject, eventdata, handles)

% hObject handle to axes1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

axes(handles.axes1);%задание текущего графика axes1

Coord = get(hObject, 'CurrentPoint');%

Получение матрицы координат указателя мышки в момент нажатия на графике

x1 = Coord(2, 1);

y1 = Coord(2, 2);

%Вывод значения параметра r в поле txtWin

strcode = num2str(x1);

set(handles.txtWin,'String',strcode);

plot(x1,y1,'or')%Построение графика

axes(handles.axes2);%задание текущего графика axes1

%Построение итерационной диаграммы

N=10;

r=x1;%Значение лямбда, которое получаем при щелчке мышкой

a=-0.36;%Начальная итерация

x=-1.5:0.1:1.5;%Вектор значений х

p=plot(x,x,'EraseMode','none'); axis([-1.5,1.5,-1,1.3])

set(p,'Xdata',x,'Ydata',logist(x,r),'color','k'),drawnow%

Построение параболы

set(p,'Xdata',x,'Ydata',x),drawnow%Построение прямой у=х

b=logist(a,r);

y=[-1,b];

x=[a,a];

set(p,'Xdata',x,'Ydata',y,'color','r'),drawnow %

Построение 1-ой вертикальной линии

y=[b,b];

x=[a,b];

set(p,'Xdata',x,'Ydata',y),drawnow %построение 1-ой горизонтальной линии

a=b;

for i=1:(N-1)

b=logist(a,r);

y=[a,b];

x=[a,a];

set(p,'Xdata',x,'Ydata',y,'color','r'),drawnow%

Построение i+1-ой вертикальной линии

y=[b,b];

x=[a,b];

set(p,'Xdata',x,'Ydata',y),drawnow%построение i+1-ой горизонтальной линии

a=b;

end

% --- Executes during object creation, after setting all properties.

function txtWin\_CreateFcn(hObject, eventdata, handles)

% hObject handle to txtWin (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

подфункция

function xdot=logist(t,r)

xdot=1-r.\*t.\*t;

