**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6**

ИССЛЕДОВАНИЕ ДИСКРЕТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

ЦЕЛЬ РАБОТЫ

Познакомиться с хаотическими свойствами простых нелинейных систем. Исследовать при помощи паутинных и бифуркационных диаграмм хаотические свойства нелинейных дискретных отображений.

ВВЕДЕНИЕ

Данное практическое занятие посвящено исследованию явления в нелинейных системах, получившего название детерминированного хаоса[[1]](#footnote-1). Этот вид хаоса порождается не случайным поведением большого количества элементов системы, а внутренней сущностью нелинейных процессов.

Одной из простейших задач, демонстрирующей хаотичность поведения системы является нелинейный рост популяции, описываемый логистическим уравнением

.

На этом примере мы познакомимся с хаотическим поведением нелинейной динамической системы. Независимо от конкретной физической природы любой объект является *динамической системой,* если можно указать такой набор величин (называемых *динамическими переменными* и характеризующих состояние системы), значения которых в любой последующий момент времени выводятся из начального набора по определенному правилу – закону эволюции.

## Дискретные отображения

Рассмотрим функцию, отображающую некоторое множество само с себя:

.

Итерация *f* (n) функции *f* определяется как композиция *f* с самой собой *n* раз:

, .

Итерацию можно выразить другим способом: .

Поскольку каждая точка  под действием итераций функции *f* перемещается по множеству *М*, то функция *f* задает дискретную динамическую систему, т.е. некое движение на множестве *М* с течением дискретного времени *n*. Если для некоторой точки определены все итерации , то множество  называется *орбитой* точки *x* под действием функции *f*.

Точка *x* называется *неподвижной точкой* функции *f*, если  для любого *n*. Неподвижная точка *x* функции *f* называется притягивающей, если все точки из некоторой ее окрестности стремятся к *x* под действием итераций функции *f*; и отталкивающей, если все точки из некоторой окрестности покидают эту окрестность под действием итераций. Если неподвижная точка является притягивающей либо отталкивающей, то она также называется гиперболической.

Точка *x* называется *периодической точкой* функции *f* периода *k*, если , причем  при *i < k*. Орбита периодической точки состоит из *k* точек и называется циклом периода *k*. Точка *x* является периодической точкой функции *f* периода *k*, если она является неподвижной точкой итерации , но не является периодической точкой итераций с меньшими номерами.

Определив основные термины, можно вернуться к исследованию логистической функции. Исследуемая функция является функцией одного параметра *λ*. Наша задача выяснить, как меняются орбиты точек, при изменении параметра *λ*. Чаше всего, при малом изменении параметра динамика, определяемая функцией, меняется мало. Немного изменяются значения неподвижных, периодических точек. Однако при некоторых значениях параметра происходит резкое изменение качественной картины, например, изменяется количество неподвижных точек, меняется их характер (притягивающие превращаются в отталкивающие). Скачкообразное изменение качественного поведения системы при плавном изменении параметра называется бифуркацией.

Рассмотрим, что будет происходить с точками из интервала [0;1] под действием итераций *логистической функции* .

При 0 < *λ* < 1 точка *x* = 0 является притягивающей, и каждая точка выбранного отрезка под действием итераций стремится к ней. При *λ* = 1 происходит бифуркация, и точка *x* = 0 перестает быть притягивающей.

При 1 < *λ* < 3 точка *x* = 0 является отталкивающей, а точка – притягивающей неподвижной точкой.

Следующая бифуркация происходит при *λ* = 3, когда точка перестает быть притягивающей и превращается в отталкивающую при *λ* > 3. Поскольку теперь обе точки *x1* и *x2* являются отталкивающими, логично предположить о наличии некого притягивающего объекта между ними. Так оно и происходит. При *λ* > 3 появляются еще две неподвижные точки, которые не являются неподвижными точками исходной функции. Они образуют цикл периода 2. Таким образом, имеет место бифуркация удвоения периода.

При дальнейшем увеличении параметра *λ* можно наблюдать и дальнейшие бифуркации удвоения периода. Для того чтобы наблюдать весь этот каскад бифуркаций можно провести следующий эксперимент. Выберем какое-либо начальное значение *x* и проделаем над ним 100-200 итераций отображения логистической функции. Затем отложим значения следующих 100-200 итераций по вертикальной оси, а значение параметра *λ*, при котором производились вычисления, – по горизонтальной. По оси *λ* пройдем отрезок от 2,9 до 4 с небольшим интервалом, например 0,01. Полученное множество называется бифуркационной диаграммой логистической функции.

Бифуркационная диаграмма (рис. 1) позволяет проследить за развитием системы при плавном изменении параметра. При фиксированном значении параметра за орбитами точек позволяет проследить паутинная диаграмма (диаграмма Ламерея), изображенная на рис. 2. Построение паутинной диаграммы позволяет выявить различные эффекты, незаметные на бифуркационной.

## Порядок построения паутинной диаграммы

1. Стоится график исследуемой функции. В нашем случае – .

2. Строим прямую .

3. Строим «траекторию» – ломанную, которая исходит из начального значения *x*, затем отражается под прямым углом от графика исследуемой функции, отражается от прямой , потом отражается от графика функции и т.д. Действие этого правила легко пронаблюдать, если внимательно проследить за процессом построения паутинной диаграммы при различных значениях параметра *λ.*

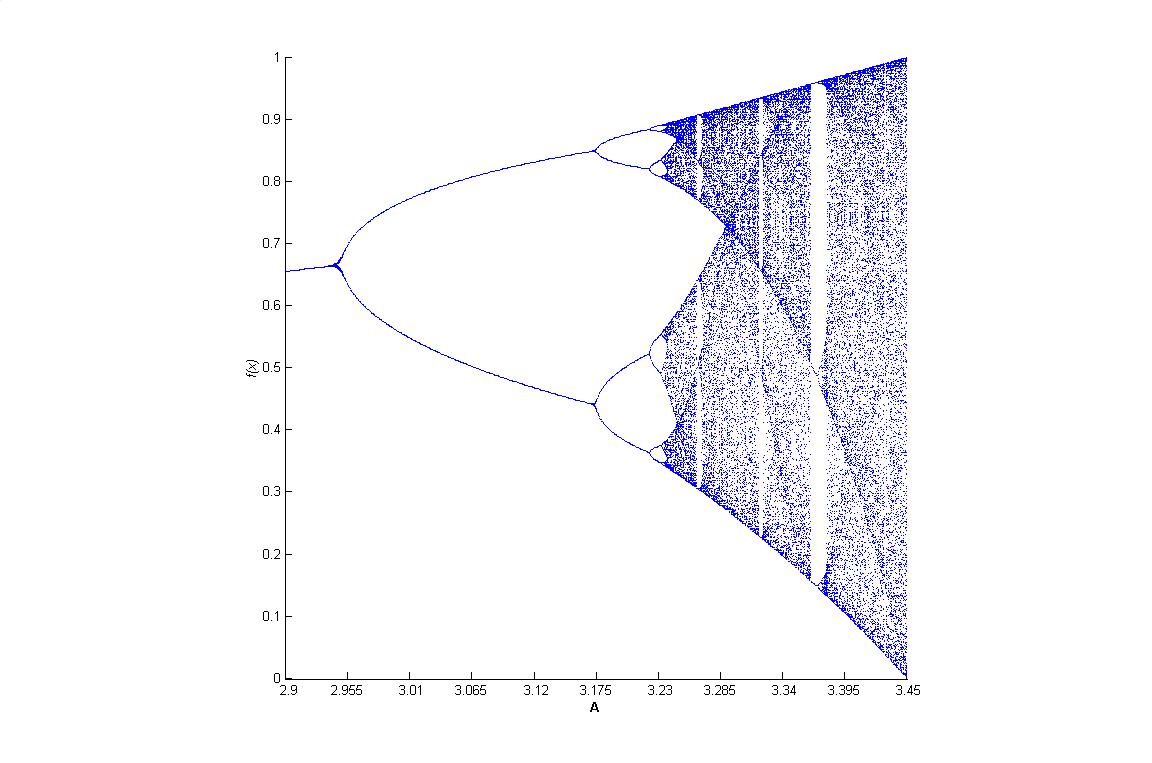


Рис. 1. Бифуркационная диаграмма логистической функции.

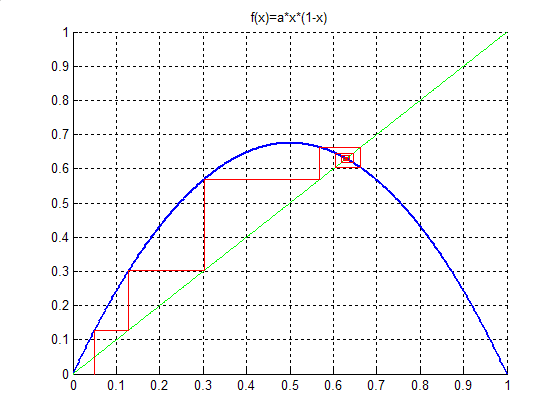


Рис. 2. Паутинная диаграмма логистической функции при *λ* = 2.7

## Порядок построения бифуркационной диаграммы

1. Выбираем начальное значение *x* (например, *x =* 0.5) и начальное значение *λ*.

2. Производим 200 итераций заданного отображения.

3. Запоминаем или отображаем значения последующих 200 итераций отображения.

4. Увеличиваем значение *λ* на заданный шаг и повторяем процедуру вычислений.

**ЗАДАНИЕ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ**

1. Ознакомиться с теоретическим введением к практическому занятию.
2. При помощи программы *Chaos* (в среде *MatLab*) построить паутинные диаграммы нелинейных дискретных отображений в соответствии со своим вариантом при различных значениях параметра *λ*. Продемонстрировать периодические и хаотические режимы.
3. При помощи программы *Chaos* (в среде *MatLab*) построить бифуркационные диаграммы нелинейных дискретных отображений в соответствии со своим вариантом.
4. Объяснить и продемонстрировать связь между паутинной и бифуркационной диаграммой.
5. Написать функцию, позволяющую построить бифуркационную диаграмму для заданного нелинейного дискретного отображения.
6. Сравнить полученные результаты с результатами программы *Chaos*.

## Варианты заданий

|  |  |  |
| --- | --- | --- |
| №  варианта | Функция | «Рабочий» диапазон  параметра *λ* |
| 1 |  | 1 – 4 |
| 2 |  | 0.9 – 2 |
| 3 |  | 1 – 3 |
| 4 |  | 0 – 2 |

Для каждого варианта имеются сохраненные настройки программы *Chaos* (файл *var#.chs*).

## Пояснения к выполнению практического занятия

1. При работе с программой *Chaos* необходимо помнить, что параметр функции *λ* обозначен как *a*.

2. В поле «Исследуемая функция» задается функция в соответствии правилами, принятыми в среде *MatLab*. В связи с этим многие операции принудительно заданы как поэлементные («.+», «.\*»). В файле с Вашим вариантом задана правильная запись исследуемой функции, изменять эту запись не рекомендуется.

3. Для сохранения результатов построения паутинной диаграммы необходимо выполнить следующее: установить задержку рисования в 0; выбрать «Команды -> Построить паутинную диаграмму в новом окне». В этом окне можно будет сохранить результаты с помощью стандартных средств *MatLab*.

4. При написании программы для построения паутинной диаграммы рекомендуется сделать 2 варианта: первый – построение диаграммы при помощи функции plot; второй – анимационное построение при помощи функции mycomet (функция аналогична стандартной функции comet, но в качестве 4-го параметра получает значение задержки в секундах между шагами).

5. Наиболее простым и эффективным способом построение бифуркационной диаграммы является следующий: для хранения результатов заводится специальный массив, размерностью КоличествоТочекПоХ × КоличествоТочекПоА (КоличествоТочекПоА вычисляется очень просто: (МаксимальноеЗначениеА – МинимальноеЗначениеА) / ШагПоА + 1; КоличествоТочекПоХ вычисляется аналогично); затем результаты итераций (после пропущенных 100) «регистрируются» в этом массиве; после окончания всех вычислений массив выводится на экран при помощи функции spy().

### «Регистрация точек в массиве»

Рассмотрим пример. Мы выбрали в качестве исследуемого отображения логистическую функцию, , , шаг по *λ* выбран 0.2. Таким образом КоличествоТочекПоА = 5. КоличествоТочекПоХ примем также равное 5. Исходный массив результата имеет вид:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Начнем выполнять итерации при λ = 3. На каком-то регистрируемом шаге мы получили значение *x* = 0.35. Необходимо занести это значение в массив. Представим, что весь интервал  разбит на 5 интервалов  – это и есть столбец нашего массива результатов. Значение *x* = 0.35 попадает во второй интервал (int(0.35/0.2) + 1 = 2), следовательно, во второй строке должна появиться отметка об этом. Проще всего записывать 1. Проведя аналогичные рассуждения относительно λ, получаем, что 1 должна записаться в 1 столбец, т.е.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Аналогично выполняется «регистрация» результатов последующих итераций.

6. Необходимо помнить, что MatLAB оптимизирован для выполнения векторных операций. Т.е. код x = [0:0.1:1]; y = sin(x); будет выполняться значительно быстрее, чем соответствующий перебор массива *x* и вычисления для него массива *y*.

7. Инициализацию массива удобно выполнять при помощи команды zeros(m,n).

**ЛИТЕРАТУРА К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ № 6**

1. Медведева Н.Б. Динамика логистической функции. Соросовский образовательный журнал, том 6, №8, 2000.
2. Климонтович Ю.Л. Введение в физику открытых систем. Соросовский образовательный журнал, №8, 1996.
3. Кузнецов А.П. Наглядные образы хаоса. Соросовский образовательный журнал, том 6, №11, 2000.
4. Кузнецов С.П. Динамический хаос. – М: Физматлит, 2001.

1. Кроме термина «детерминированный хаос» часто встречается термин «динамический хаос». [↑](#footnote-ref-1)