

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Математики и информационных технологий
Кафедра Информационных технологий

Работа допущена к защите

Зав.кафедрой Волков М. А.
(Ф. и. О)

(подпись)

(дата)

ДИПЛОМНАЯ РАБОТА

Средства электронной поддержки курса «Основы моделирования»

(название темы)

Прикладная математика и информатика. 010501

(наименование и номер специальности)

Проект выполнил студент ПМ-51 _____ Уресметова И. В.
(группа) (подпись) (Ф.И.О.)

Руководитель профессор _____ Семушин И. В.
(должность) (подпись) (Ф.И.О.)

Рецензент доцент _____ Сибатов Р. Т.
(должность) (подпись) (Ф.И.О.)

**Ульяновск
2007г.**

Оглавление

Введение	3
1 Модели и моделирование	5
1.1 Виды моделей и моделирования	5
1.2 Построение математических моделей	8
1.2.1 Динамическая система и ее математическая модель	11
1.2.2 Модели фракталов	27
2 Компьютерная реализация математических моделей	30
2.1 Среда разработки средства электронной поддержки курса «Основы моделирования»	31
2.2 Назначение, структура и возможности программы	33
Заключение	47
Библиографический список	49
Приложения	52
A Листинг программы BaseOfModelling	52
B Таблица преобразований Лапласа	62

Введение

Моделирование (в широком смысле) является основным методом исследований во всех областях знаний и научно-обоснованным методом оценок характеристик сложных систем, используемым для принятия решений в различных сферах инженерной деятельности. Существующие и проектируемые системы можно эффективно исследовать с помощью математических моделей (аналитических и имитационных), решаемых на современных ЭВМ, которые в этом случае выступают в качестве инструмента экспериментатора с моделью системы.

В последние десятилетия математическое моделирование оформилось в отдельную междисциплинарную область знаний с присущими ей объектами, подходами и методами исследования. Появилось множество специальных пакетов моделирования, но все они разработаны для профессионалов в области моделирования. В связи с этим все более **актуальной** становится задача целенаправленной подготовки специалистов-матмодельеров в вузах различного профиля, в рамках различных направлений и специальностей, например, курса «Основы моделирования».

Объект исследования дипломной работы: курс «Основы моделирования», читаемый на первом курсе Факультета Математики и Информационных технологий УлГУ студентам математических специальностей с начальной подготовкой в области программирования и начальным уровнем владения математическим аппаратом.

Цель дипломной работы: разработка программно-методической поддержки курса «Основы моделирования».

В соответствии с объектом и целью работы предусматривается реше-

ние следующих **задач**:

- Разработать компьютерные программы для среды Matlab, моделирующих заданные математические модели;
- Разработать руководство пользователя к этим компьютерным программам, включая краткие инструкции по написанию m-файлов и их запуску в среде Matlab;
- Разработать варианты практических заданий по курсу «Основы моделирования», предлагаемые студентам для изучения.

Теоретическую и методологическую основу исследования составляют научные труды отечественных и зарубежных авторов в области математического моделирования: Горстко А. Б. [8], Самарский А. А. [20], E. R. Scheinerman [29], построения фракталов: Мандельброт Б. [15], Федер Е. [23]. В процессе проведения настоящего исследования использовались следующие **методы**: анализ документации и вычислительный эксперимент.

Практическая значимость работы заключается в том, что разработанная программа может использоваться на практических (лабораторных) занятиях по курсу «Основы моделирования».

Достоверность данных обеспечивается разнообразием источников информации, исследовательских процедур и приемов, использованием методик, адекватных предмету и задачам работы, и подтверждается сочетанием количественного и качественного анализов результатов специальных численных экспериментов, результаты которых априорно известны.

Глава 1

Модели и моделирование

1.1 Виды моделей и моделирования

Моделью объекта называется любой другой объект, отдельные свойства которого полностью или частично совпадают со свойствами исходного. Процесс построения модели называется **моделированием**. Модель не может быть исчерпывающе полной, она всегда ограничена и должна лишь соответствовать целям моделирования, отражая ровно столько свойств исходного объекта и в такой полноте, сколько необходимо для конкретного исследования.

Исходный объект может быть либо реальным, либо воображаемым. С воображаемыми объектами в инженерной практике имеют дело на ранних этапах проектирования технических систем. Модели еще не воплощенных в реальные разработки объектов называются *предвосхищающими*.

Модель создается ради исследований, которые на реальном объекте проводить либо невозможно, либо дорого, либо просто неудобно. Можно выделить несколько *целей*, ради которых создаются модели и ряд основных *типов исследований* [21]:

- модель как средство осмысления помогает выявить взаимозависимости переменных, характер их изменения во времени, найти существующие закономерности. При составлении модели становится более понятной структура исследуемого объекта, вскрываются важ-

ные причинно-следственные связи. В процессе моделирования постепенно происходит разделение свойств исходного объекта на существенные и второстепенные с точки зрения сформулированных требований к модели. Пытаются найти в исходном объекте только те черты, которые имеют непосредственное отношение к интересующей нас стороне его функционирования. В определенном смысле вся научная деятельность сводится к построению и исследованию моделей *природных явлений*;

- модель как средство прогнозирования позволяет научиться предсказывать поведение объекта и управлять им, испытывая различные варианты управления на модели. Экспериментировать с реальным объектом часто, в лучшем случае, бывает неудобно, а иногда и просто опасно или вообще невозможно в силу ряда причин: большой продолжительности эксперимента, риска повредить или уничтожить объект, отсутствия реального объекта в случае, когда он еще только проектируется;
- построенные модели могут использоваться для нахождения оптимальных соотношений параметров, исследования *особых (критических) режимов работы*;
- модель также может в некоторых случаях заменять исходный объект при обучении, например, использоваться в качестве тренажера при подготовке персонала к последующей работе в реальной обстановке, или выступать в качестве исследуемого объекта в виртуальной лаборатории. Модели, реализованные в виде исполняемых модулей, применяются и как имитаторы объектов управления при стендовых испытаниях систем управления, и, на ранних стадиях проектирования, заменяют сами будущие аппаратно реализуемые системы управления.

Модели можно условно разделить на две группы: *материальные и идеальные*, и, соответственно, различать предметное и абстрактное мо-

делирование. Основными разновидностями предметного моделирования являются физическое и аналоговое моделирование [5].

Физическим принято называть такое моделирование (макетирование), при котором реальному объекту ставится в соответствие его увеличенная или уменьшенная копия. Эта копия создается на основе теории подобия, что и позволяет утверждать, что в модели сохранились требуемые свойства. В физических моделях помимо геометрических пропорций может быть сохранен, например, материал или цветовая гамма исходного объекта, а также другие свойства, необходимые для конкретного исследования.

Аналоговое моделирование основано на замене исходного объекта объектом другой физической природы, обладающим аналогичным поведением. При аналоговом моделировании важно увидеть в объекте-заменителе нужные черты, и правильно их интерпретировать.

От предметного моделирования принципиально отличается идеальное моделирование, которое основано не на математической аналогии, а на аналогии идеальной, мыслимой. Различают два типа идеального моделирования: интуитивное и знаковое.

Под *интуитивным* понимается моделирование, основанное на интуитивном представлении об объекте исследования, не поддающемся формализации либо не нуждающемся в ней.

Знаковым называется моделирование, использующее в качестве моделей знаковые преобразования какого-либо вида: схемы, графики, чертежи, формулы, наборы символов и т. д., а также включающие совокупность законов, по которым можно оперировать с выбранными знаковыми образованиями и их элементами.

Важнейшим видом знакового моделирования является **математическое моделирование**, при котором исследование объекта осуществляется посредством модели, сформулированной на языке математики или иных математических методов.

1.2 Построение математических моделей

Бурное развитие методов математического моделирования и многообразии областей их использования привело к появлению огромного количества моделей самого разного типа [3].

Любая математическая модель может рассматриваться как некоторый оператор, который является алгоритмом или определяется совокупностью уравнений — алгебраических, обыкновенных дифференциальных уравнений, систем обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, интегро-дифференциальных уравнений и др.

Исторически первыми стали разрабатываться и исследоваться *линейные* математические модели. Область применения подобных моделей очень широка. Она охватывает (но не ограничивается этим) классическую механику, электродинамику, аналитическую химию и биологию. Методы их построения, разрабатывавшиеся в течение столетий, обладают большой общностью и эффективностью.

В настоящее время все чаще возникает потребность не только в повышении точности моделирования, но и в создании качественно новых моделей, учитывающих нелинейность поведения реальных объектов исследования [22]. Анализ подобных моделей намного сложнее, чем линейных, причем разработка методики и общих подходов к исследованию в настоящее время далека от завершения. Являясь более богатым и сложным, мир *нелинейных* моделей представляется для современной науки более перспективным в плане открытия новых закономерностей и описания сложных явлений.

В зависимости от вида используемых при их создании множеств модели могут подразделяться на *качественные* и *количественные*, *дискретные* и *непрерывные*, а также *смешанные*.

При построении моделей реальных объектов и явлений очень часто приходится сталкиваться с недостатком информации. Как правило, для

любого исследуемого объекта распределение свойств, параметры воздействия и начальное состояние известны с той или иной степенью неопределенности. Это связано с множеством трудно учитываемых факторов, ограниченностью числа используемых параметров модели, конечной точностью экспериментальных измерений. При построении модели возможны следующие варианты описания неопределенности параметров: *детерминированное, стохастическое, случайное, интервальное* или *нечеткое*.

Разделение моделей на *одномерные, двумерные* и *трехмерные* применимо для таких моделей, в число параметров которых входят координаты пространства, и связано с особенностями реализации этих моделей, равно как и с резким увеличением их сложности при возрастании размерности. Как правило, увеличение размерности модели приводит к росту числа используемых математических соотношений. Особенно сложны в реализации трехмерные модели, требующие высокопроизводительной вычислительной техники с большим объемом оперативной и дисковой памяти. Реализация таких моделей стала возможной лишь с появлением вычислительных машин третьего поколения и потребовала создания специальных вычислительных методов и приемов.

Из всей совокупности параметров при разработке различных моделей отдельно следует рассмотреть *учет времени*. Учитывая, что весь окружающий нас материальный мир состоит из постоянно изменяющихся и взаимодействующих элементарных частиц и полей, то все без исключения объекты исследования из этого мира следует считать изменяющимися во времени.

Любой объект стремится перейти в некоторое *равновесное состояние* как с окружающей его средой, так и между отдельными элементами самого объекта. Нарушение этого равновесия приводит к изменениям различных параметров объекта и его переходу в новое равновесное состояние.

При построении модели важным является сравнение времени существенных изменений внешних воздействий и характерных времен перехо-

да объекта в новое равновесное состояние с окружающей средой, а также времени релаксации, определяющего установление равновесия между отдельными элементами внутри объекта. Если скорости изменения внешних воздействий на объект моделирования существенно меньше скорости релаксации, то явной зависимостью от времени в модели можно пренебречь. В этом случае говорят о *квазистатическом процессе*.

Совокупность значений параметров модели в некоторый момент времени или на данной стадии называется *состоянием объекта*.

Если скорости изменения внешних воздействий и параметров состояния изучаемого объекта достаточно велики (по сравнению со скоростями релаксации), то учет времени необходим. В этом случае объект исследования рассматривают в рамках *динамического процесса*.

В случае, когда в каждой фиксированной точке исследуемого пространства значения параметров модели не зависят от времени, говорят о *стационарных процессах*. Как правило, стационарные модели применяются для описания различных потоков (жидкости, газа, тепла) в случае постоянства условий на входе и выходе потока. Для таких процессов время может быть исключено из числа независимых переменных.

Если в качестве одной из существенных независимых переменных модели необходимо использовать время (или его аналог), то модель называется *нестационарной*.

В данной работе не охватываются все виды моделей. Мы ограничимся рассмотрением динамических систем [1] и фракталов [17, 12]. Подобный выбор обусловлен ориентированностью дипломной работы на курс «Основы моделирования».

Данный курс читается для студентов факультета Математики и информационных технологий УлГУ в первом семестре. Цели курса «Основы моделирования»:

- передать студентам определенную систему знаний, умений, навыков;
- помочь студентам овладеть математическими методами познания

реальной действительности;

- помочь студентам овладеть минимумом математических сведений для того, чтобы применять эти сведения;
- развитие логического мышления;
- развитие элементов творческой деятельности как качеств мышления (интуиция, воображение, смекалка);
- развитие мировоззрения;
- воспитание устойчивого интереса к изучению математики;
- воспитание патриотизма и эмоционального самосознания;
- эстетическое воспитание, воспитание общей культуры;
- ознакомление с ролью математики в развитии культуры и науки;

1.2.1 Динамическая система и ее математическая модель

Под **динамической системой** понимают любой объект или процесс, для которого однозначно определено понятие состояния как совокупности некоторых величин в данный момент времени и задан закон, который описывает изменение (эволюцию) начального состояния с течением времени [29]. Этот закон позволяет по начальному состоянию прогнозировать будущее состояние динамической системы, его называют законом эволюции. Она может иметь как *линейную* структуру, так и *нелинейную*. Есть два вида динамических систем: *дискретные во времени* и *непрерывные во времени*.

Для дискретных систем вводится дискретное время k , тогда система определяется уравнениями: $x(k+1) = f(x(k))$ и $x(0) = x_0$

Для непрерывного времени вводится время t , тогда система задается следующими уравнениями: $\dot{x} = f(x)$ и $x(0) = x_0$, где \dot{x} — производная по времени и $f(x)$ — некоторая функция.

Динамические системы — это механические, физические, химические и биологические объекты, вычислительные процессы и процессы преобразования информации, совершаемые в соответствии с конкретными алгоритмами. Описания динамических систем для задания закона эволюции также разнообразны: с помощью дифференциальных уравнений, дискретных отображений, теории графов, теории марковских цепей и т. д..

Учитывая, что студенты еще не владеют аппаратом математического анализа, алгебры, дифференциальных уравнений и других математических дисциплин, (согласно принципам доступности, системности и последовательности) в данной работе рассмотрены алгебраические уравнения или системы, заданные обыкновенными дифференциальными уравнениями, размерности ≤ 4 .

Математическая модель динамической системы считается заданной, если введены параметры (координаты) системы, определяющие однозначно ее состояние, и указан закон эволюции. В зависимости от степени приближения одной и той же системе могут быть поставлены в соответствие различные математические модели. Исследование реальных систем сводится к изучению математических моделей, совершенствование и развитие которых определяются анализом экспериментальных и теоретических результатов при их сопоставлении. В связи с этим под динамической системой мы будем понимать именно ее математическую модель.

Практически все приведенные здесь результаты могут быть получены студентами самостоятельно.

ЗАДАЧА 1: Математический маятник

Общеизвестным примером колебательной механической системы является «математический» маятник (рис. 1.1). Это идеализированная система, состоящая из частицы или груза массой m , прикрепленной к нижнему концу жесткого стержня длиной L с пренебрежимо малой массой, верхний конец которого вращается без трения в точке подвеса. Если груз

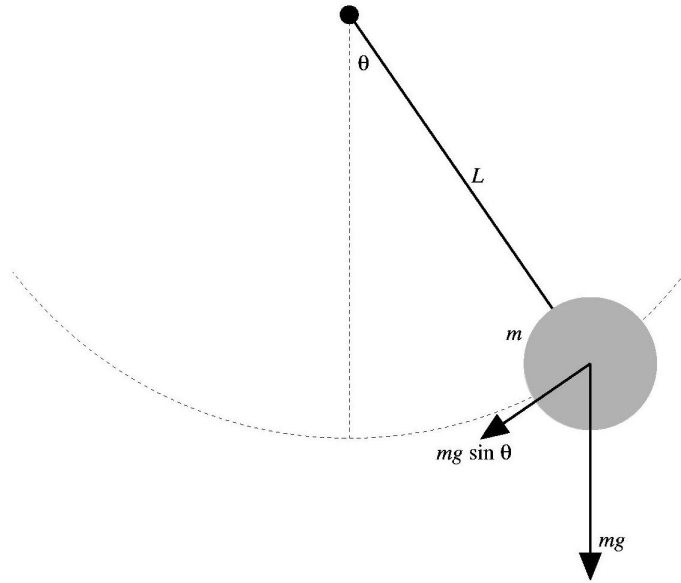


Рис. 1.1: Силы, действующие на математический маятник. Угол θ измеряется относительно вертикальной оси. Он положителен, когда груз находится справа, и отрицателен, когда груз находится слева.

вывести из положения равновесия и отпустить, то маятник будет совершать колебания в вертикальной плоскости. Необходимо получить представление о движении маятника в случае малых амплитуд колебаний.

РЕШЕНИЕ: Поскольку движение груза происходит по дуге окружности радиуса L с центром в точке 0 , то положение груза характеризуется длиной дуги, или углом θ . Угловая скорость ω и ускорение груза a при движении по дуге равны

$$\omega = L \frac{d\theta}{dt}$$

$$a = L \frac{d^2\theta}{dt^2}$$

В отсутствии трения на тело действуют две силы: сила тяжести $F = mg$, направленная вертикально вниз, и сила со стороны стержня. Последняя направлена из центра, если $|\theta| < \pi/2$. Поскольку стержень жесткий, то необходимо учесть только компоненту силы mg , направленную по касательной к дуге. Эта компонента (см. рис. 1.1) равна $mg \sin \theta$ и направлена в сторону уменьшения угла θ . Тогда уравнение движения

запишется в виде

$$mL \frac{d^2\theta}{dt^2} = -mg \sin \theta$$

или

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L} \sin \theta \quad (1.1)$$

Уравнение (1.1) является примером нелинейного уравнения, поскольку в него входит $\sin \theta$, а не θ . Большинство нелинейных уравнений не имеет аналитических решений в элементарных функциях, и (1.1) не является исключением. Однако если амплитуда колебаний маятника достаточно мала, то (1.1) можно переписать в виде

$$\frac{d^2\theta}{dt^2} = -\theta \frac{g}{L} \quad (1.2)$$

для $\theta \ll 1$. (Напомним, что θ измеряется в радианах.)

Обозначим $\omega_n^2 = \frac{g}{L}$, где n обозначает «natural», т. е. собственная частота.

Перепишем (1.2) в виде

$$\ddot{\theta}(t) + \omega_n^2 \theta(t) = 0 \quad (1.3)$$

при начальных условиях

$$\theta(0) = \theta_0 \quad (1.4)$$

$$\dot{\theta}(0) = \omega_0$$

Способ 1:

Найдем решение через преобразования Лапласа (в частотной области), учитывая правило нахождения изображений для производных

$$\theta(t) \div X(s) \quad (1.5)$$

$$\dot{\theta}(t) \div sX(s) - \theta_0 \quad (1.6)$$

$$\ddot{\theta}(t) \div s^2 X(s) - s\theta_0 - \omega_0 \quad (1.7)$$

Здесь слева записаны «оригиналы», а справа от \div — их «изображения» по Лапласу.

Тогда наше уравнение (1.3) можно переписать в терминах изображений:

$$\begin{aligned} [s^2 X(s) - s\theta_0 - \omega_0] + \omega_n^2 X(s) &= 0 \\ (s^2 + \omega_n^2) X(s) &= s\theta_0 + \omega_0 \end{aligned}$$

$$X(s) = \frac{s}{s^2 + \omega_n^2} \theta_0 + \frac{\omega_n}{s^2 + \omega_n^2} \frac{\omega_0}{\omega_n}$$

Используя табл. В.1(Приложение В), нахождения оригинала для дробно-рациональных изображений, получим

$$\theta(t) = \theta_0 \cos \omega_n t + \frac{\omega_0}{\omega_n} \sin \omega_n t \quad (1.8)$$

Преобразуя (1.8) по формуле синуса суммы:

$$\theta(t) = \sqrt{\theta_0^2 + \left(\frac{\omega_0}{\omega_n}\right)^2} \left[\frac{\theta_0}{\sqrt{\theta_0^2 + \left(\frac{\omega_0}{\omega_n}\right)^2}} \cos \omega_n t + \frac{\frac{\omega_0}{\omega_n}}{\sqrt{\theta_0^2 + \left(\frac{\omega_0}{\omega_n}\right)^2}} \sin \omega_n t \right]$$

имеем окончательное решение

$$\theta(t) = A \sin(\omega_n t + \alpha) \quad (1.9)$$

где

$$\begin{aligned} A &= \sqrt{\theta_0^2 + \left(\frac{\omega_0}{\omega_n}\right)^2} \\ \operatorname{tg} \alpha &= \frac{\theta_0}{\left(\frac{\omega_0}{\omega_n}\right)} \end{aligned}$$

Способ 2:

Найдем решение в пространстве состояний (во временной области)

Обозначим $x = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$. Тогда уравнение (1.3) перепишем в виде

системы $\dot{x}(t) = Fx(t)$, где $F = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & 0 \end{bmatrix}$ и $x_0 = \begin{bmatrix} \theta_0 \\ \omega_0 \end{bmatrix}$

Используя соответствия (1.5), получим

$$\begin{aligned} sX(s) - x_0 &= FX(s) \\ (Is - F)X(s) &= x_0 \\ X(s) &= (Is - F)^{-1}x_0 \end{aligned}$$

Найдем значение матрицы $(Is - F)^{-1}$

$$(Is - F)^{-1} = \begin{bmatrix} s & -1 \\ \omega_n^2 & s \end{bmatrix}^{-1} = \begin{bmatrix} \frac{s}{s^2 + \omega_n^2} & \frac{1}{s^2 + \omega_n^2} \\ -\frac{\omega_n^2}{s^2 + \omega_n^2} & \frac{s}{s^2 + \omega_n^2} \end{bmatrix}$$

Используя табл. В.1(Приложение В), нахождения оригинала для дробно-рациональных изображений, получим

$$\begin{bmatrix} \cos(\omega_n t) & \frac{1}{\omega_n} \sin(\omega_n t) \\ -\omega_n \sin(\omega_n t) & \cos(\omega_n t) \end{bmatrix}$$

Другими словами,

$$x(t) = \begin{bmatrix} \cos(\omega_n t) & \frac{1}{\omega_n} \sin(\omega_n t) \\ -\omega_n \sin(\omega_n t) & \cos(\omega_n t) \end{bmatrix} x_0 = \begin{bmatrix} \cos(\omega_n t) \theta_0 + \sin(\omega_n t) \frac{\omega_0}{\omega_n} \\ -\omega_n \sin(\omega_n t) \theta_0 + \cos(\omega_n t) \omega_0 \end{bmatrix}$$

Подставляя различные значения параметров (все рисунки получены в результате работы программы BaseOfModelling, глава 2), можно получить теперь траекторию движения физического маятника (рис. 1.2, 1.3, 1.4).

ЗАДАЧА 2: Простейшая модель «хищник - жертва»

Пусть есть два биологических вида, которые совместно обитают в изолированной среде. Среда стационарна и обеспечивает в неограниченном количестве всем необходимым для жизни один из видов, который будем называть жертвой. Другой вид — хищник также находится в стационарных условиях, но питается лишь особями первого вида.

Это могут быть караси и щуки, зайцы и волки, мыши и лисы, микробы и антитела и т. д.. Будем для определенности называть их карасями и щуками.

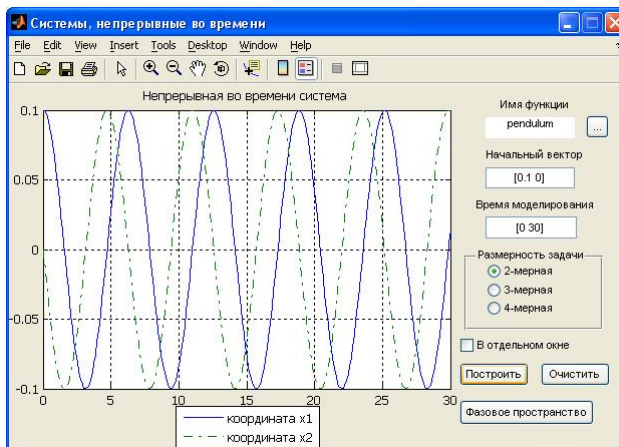


Рис. 1.2: Траектория движения физического маятника при $\theta_0 = 0.1$ и $\omega_0 = 0$. Непрерывная линия это θ , пунктирная ω . Горизонтальная ось - время t

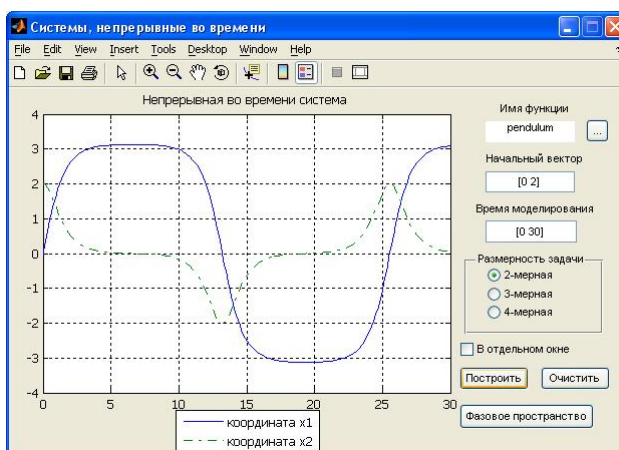


Рис. 1.3: Траектория движения физического маятника при $\theta_0 = 0$ и $\omega_0 = 2$. Непрерывная линия это θ , пунктирная ω . Горизонтальная ось - время t

Караси и щуки живут в некотором изолированном пруду. Среда предоставляет карасям питание в неограниченном количестве, а щуки питаются лишь карасями.

Как будут меняться численности популяции, если сдерживающие факторы (болезни, конкурирующие виды, ограниченность питания) отсутствуют?

РЕШЕНИЕ:

Обозначим

- $x_1(t)$ - число карасей в момент t

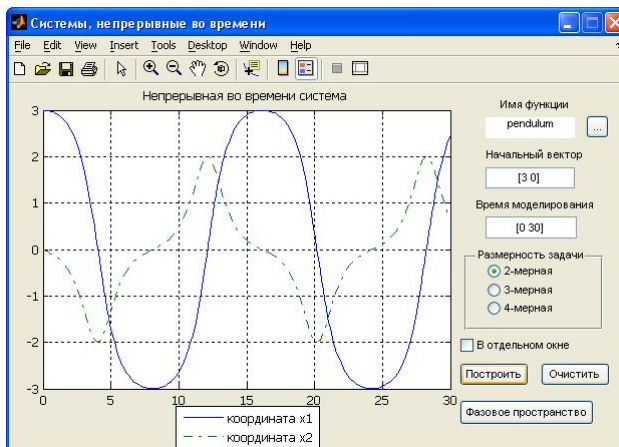


Рис. 1.4: Траектория движения физического маятника при $\theta_0 = 3$ и $\omega_0 = 0$. Непрерывная линия это θ , пунктирная ω . Горизонтальная ось - время t

- $x_2(t)$ - число щук в момент t

Будем называть пару чисел (x_1, x_2) состоянием модели. Попробуем определить, как это состояние меняется с течением времени.

Рассмотрим \dot{x}_1 — скорость изменения численности карасей. Если щук нет, то число карасей увеличивается тем быстрее, чем больше карасей. Будем считать, что эта зависимость линейная: $\dot{x}_1 \sim \varepsilon_1 x_1$, причем коэффициент ε_1 зависит только от условий жизни карасей, их естественной смертности и рождаемости.

Скорость изменения \dot{x}_2 числа щук (если нет карасей), зависит от числа щук x_2 .

Будем считать, что $\dot{x}_2 \sim -\varepsilon_2 x_2$. Если карасей нет, то число щук уменьшается (у них нет пищи) и они вымирают.

В экосистеме скорость изменения численности каждого вида также будем считать пропорциональной его численности, но только с коэффициентом, который зависит от численности особей другого вида. Так, для карасей этот коэффициент уменьшается с увеличением числа щук, а для щук увеличивается с увеличением числа карасей. Будем считать эту зависимость также линейной.

Тогда получим систему из двух дифференциальных уравнений:

$$\begin{aligned} \dot{x}_1 &= \varepsilon_1 x_1 - \gamma_1 x_1 x_2 \\ \dot{x}_2 &= -\varepsilon_2 x_2 + \gamma_2 x_1 x_2 \end{aligned} \quad (1.10)$$

Эта система уравнений называется моделью Вольтерра-Лотки. Числовые коэффициенты $\varepsilon_1, \gamma_1, \varepsilon_2, \gamma_2$ называются параметрами модели. Очевидно, что характер изменения состояния (x_1, x_2) определяется значениями этих параметров.

Перепишем эту систему в форме $\dot{x} = f(x)$, где $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ и

$$f \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \varepsilon_1 x_1 - \gamma_1 x_1 x_2 \\ -\varepsilon_2 x_2 + \gamma_2 x_1 x_2 \end{bmatrix}$$

Для дискретизации (1.10) существует немало численных методов: одношаговые, многошаговые, и т. д., например, метод Рунге-Кутты 2 порядка [6]:

$$x^{i+1} = x^i + h \left[(1 - \alpha) f(x^i) + \alpha f \left(x^i + \frac{h}{2\alpha} \right) \right]$$

где h - шаг дискретизации, $\alpha (\neq 0)$ - параметр.

Изменяя параметры и решая систему уравнений модели можно исследовать закономерности изменения состояния системы «хищник-жертва».

Пусть $\alpha = \frac{1}{2}$, $\varepsilon_1 = 0.2$, $\varepsilon_2 = 0.1$, $\gamma_1 = 0.002$, $\gamma_2 = 0.001$, $x_0 = \begin{bmatrix} 100 \\ 25 \end{bmatrix}$

В качестве примера на рис. 1.5 построены кривые изменения численности карасей и щук в зависимости от времени t для заданных значений параметров. Численность популяции карасей и щук повторяются через некоторое время. Поведение популяций является периодическим. Когда много карасей, тогда продовольствия для щук становится больше, растет популяция щук. Но это влечет за собой сверхохоту на карасей. В результате падает их численность. Продовольствия щукам становится недостаточно, и уменьшается их численность. Наконец, наступает такой

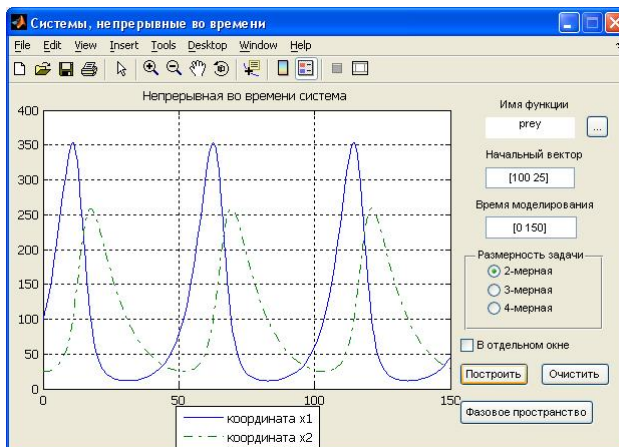


Рис. 1.5: Зависимости от времени t численности карасей x_1 и щук x_2

момент, когда численность популяции щук достаточно низкая для того, чтобы оправиться карасям и цикл повторяется снова.

Чтобы полностью оценить циклический характер этого процесса, мы можем изобразить популяции карасей и щук на отдельном графике с осью x_1 , обозначающей число карасей и осью x_2 число щук (см. рис. 1.6). Каждая точка на кривой представляет состояние системы; данную кривую принято называть фазовой диаграммой (или диаграммой фазового пространства).

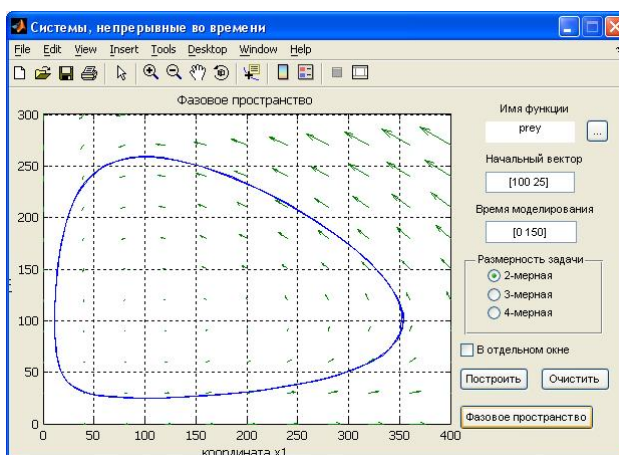


Рис. 1.6: Траектория фазового пространства задачи (1.10)

ЗАДАЧА 3: Колебательные движения

Груз весом P , подвешенный к концу пружины, движется в жидкости. Коэффициент жесткости пружины s . Сила сопротивления движению

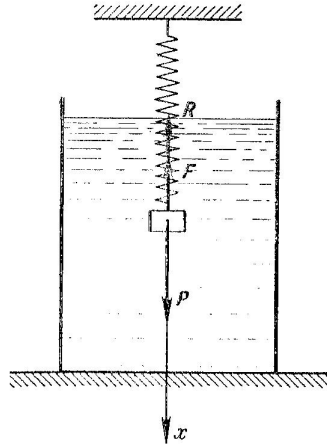


Рис. 1.7: Колебательные движения груза в жидкости и действующие на него силы пропорциональна первой степени скорости груза: $R = \gamma v$.

Найти уравнение движения груза, если в начальный момент груз был смещен из положения статического равновесия вниз на x_0 и ему была сообщена вниз начальная скорость v_0 [2].

РЕШЕНИЕ:

Направим ось x вдоль пружины по вертикали вниз, взяв начало отсчета в положении статического равновесия груза.

Изобразим груз в положении, при котором его абсцисса x положительна (рис. 1.7). Так как пружина при этом получит удлинение $\delta = \delta_{st} + x$, то проекция на ось x силы упругости F пружины, направленной по вертикали вверх, равна

$$F_x = -c(\delta_{st} + x) \quad (1.11)$$

Предположим, что груз движется в сторону возрастания абсциссы x . Сила сопротивления R движению груза направлена противоположно скорости, т.е. по вертикали вверх. Кроме сил F и R , к грузу приложен его вес $P = mg$. Составим дифференциальное уравнение движения груза:

$$m\ddot{x} = P + F_x + R_x$$

Подставив в это уравнение $R_x = -\gamma v_x$ и значение F_x из формулы (1.11), находим:

$$\frac{P}{g}\ddot{x} = P - c\delta_{st} - cx - \gamma v_x \quad (1.12)$$

Нетрудно видеть, что

$$P - c\delta_{st} = 0 \quad (1.13)$$

Действительно, в положении статического равновесия к грузу приложены следующие силы: P - его вес, направленный по вертикали вниз, и статическая сила упругости пружины $F_{st} = c\delta_{st}$, направленная по вертикали вверх. Так как груз находится в равновесии, то $P - F_{st} = 0$, откуда и следует формула (1.13).

Первые два слагаемых в правой части дифференциального уравнения (1.12), на основании формулы (1.13), уничтожаются. Заменяя в последнем слагаемом v_x на \dot{x} , имеем:

$$\ddot{x} + 2n\dot{x} + k^2x = 0, \quad (1.14)$$

где

$$k = \sqrt{\frac{cg}{P}} = \sqrt{\frac{c}{m}} \quad (1.15)$$

$$n = \frac{\gamma g}{2P} = \frac{\gamma}{2m} \quad (1.16)$$

Найдем решение через преобразования Лапласа (в частотной области)

$$x(t) \div X(s)$$

$$\dot{x}(t) \div sX(s) - x_0$$

$$\ddot{x}(t) \div s^2X(s) - sx_0 - v_0$$

Тогда наше уравнение (1.14) можно переписать:

$$s^2X(s) - sx_0 - v_0 + 2n(sX(s) - x_0) + k^2X(s) = 0$$

$$(s^2 + 2ns + k^2)X(s) = sx_0 + v_0 + 2nx_0$$

$$X(s) = \frac{sx_0 + v_0 + 2nx_0}{s^2 + 2ns + k^2}$$

Обозначим $c_0 = 2nx_0 + v_0$ и $c_1 = x_0$.

Тогда

$$X(s) = c_1 \frac{s + (c_0/c_1)}{s^2 + 2ns + k^2}$$

Пусть $\Delta = k^2 - n^2$.

Используя табл. В.1(Приложение В), нахождения оригинала для дробно-рациональных изображений, получим:

Случай 1 малого сопротивления ($\Delta > 0$)

$$\begin{aligned} x(t) &= (c_0/c_1) \frac{c_1}{\sqrt{\Delta}} e^{-nt} \sin \sqrt{\Delta} t + c_1 e^{-nt} \left(\cos(\sqrt{\Delta} t) - \frac{n}{\sqrt{\Delta}} \sin(\sqrt{\Delta} t) \right) = \\ &= \frac{1}{\sqrt{\Delta}} A e^{-nt} \sin(\sqrt{\Delta} t + \beta) \end{aligned} \quad (1.17)$$

где

$$A = \sqrt{(c_0 - nc_1)^2 + \Delta c_1^2} \quad (1.18)$$

$$\beta = \text{arctg}\left(\frac{c_1 \sqrt{\Delta}}{c_0 - nc_1}\right) \quad (1.19)$$

Внеся значения Δ, c_0, c_1 в (1.17)-(1.19), перепишем:

$$x(t) = A e^{-nt} \sin(\sqrt{k^2 - n^2} t + \beta) \quad (1.20)$$

где

$$A = \sqrt{\frac{(nx_0 + v_0)^2}{k^2 - n^2} + x_0^2} \quad (1.21)$$

$$\beta = \arctan\left(\frac{x_0 \sqrt{k^2 - n^2}}{nx_0 + v_0}\right) \quad (1.22)$$

(1.20) — уравнение движения груза.

Движение груза является затухающим (так как при $t \rightarrow \infty x \rightarrow 0$) с круговой частотой

$$k_c = \sqrt{k^2 - n^2} \quad (1.23)$$

Пусть $P = 98$ Н, $c = 10$ Н/см, $\gamma = 1$, 6нсек/см, $x_0 = 4$ см, $v_0 = 4$ см/сек. Подставив эти численные значения в формулы (1.15)-(1.16), (1.20)-(1.22), находим $k = 10$ сек⁻¹, $n = 8$ сек⁻¹, $A = 7.2$ см, $\beta = 0.59$ рад, $k_c = 6$ сек⁻¹.

Как видно из рис. 1.8, груз совершает затухающие колебания.

Случай 2 большого сопротивления ($\Delta < 0$)

$$\begin{aligned} x(t) &= \frac{c_0}{\sqrt{-\Delta}} e^{-nt} \text{sh} \sqrt{-\Delta} t + c_1 e^{-nt} \left(\text{ch}(\sqrt{-\Delta} t) - \frac{n}{\sqrt{-\Delta}} \text{sh}(\sqrt{-\Delta} t) \right) = \\ &= \frac{e^{-nt}}{\sqrt{-\Delta}} \left((c_0 - nc_1) \text{sh} \sqrt{-\Delta} t + c_1 \sqrt{-\Delta} \text{ch} \sqrt{-\Delta} t \right) \end{aligned} \quad (1.24)$$

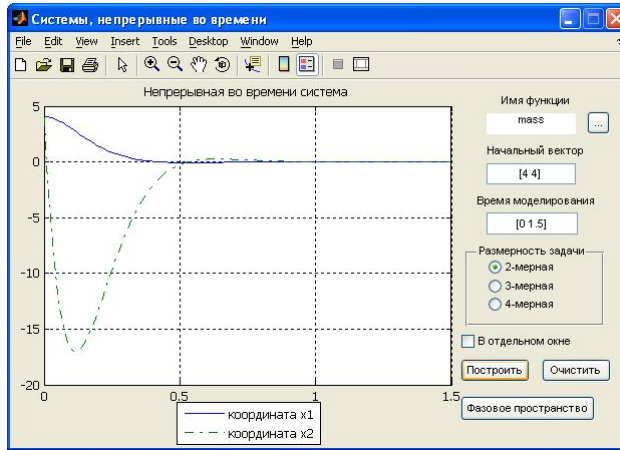


Рис. 1.8: Затухающие колебания груза в жидкости при $n < k$

Подставив в уравнение значения Δ, c_0, c_1 , находим уравнение движения груза

$$x(t) = \frac{e^{-nt}}{\sqrt{n^2 - k^2}} \left((v_0 + nx_0) \operatorname{sh} \sqrt{n^2 - k^2}t + x_0 \sqrt{n^2 - k^2} \operatorname{ch} \sqrt{n^2 - k^2}t \right) \quad (1.25)$$

Движение груза является аperiодическим и притом затухающим, так как при $t \rightarrow \infty x \rightarrow 0$.

После подстановки в уравнения (1.15)-(1.16), (1.25) численных значений $P = 98 \text{ Н}$, $c=10 \text{ Н/см}$, $\gamma = 5.2 \text{ нсек/см}$, $x_0 = 4 \text{ см}$, $v_0 = -240 \text{ см/сек}$, получим $k = 10 \text{ сек}^{-1}$, $n = 26 \text{ сек}^{-1}$ (см. рис. 1.9) и

$$x(t) = \frac{e^{-26t}}{6} (29e^{-24t} - 5e^{24t})$$

Случай 3 предельного сопротивления ($\Delta = 0, v_0 = 0$)

Уравнение движения груза имеет вид

$$x(t) = c_0 t e^{-nt} + c_1 e^{-nt} (1 - nt) \quad (1.26)$$

Подставив в уравнение (1.26) значения c_0, c_1 , получим выражение

$$x(t) = x_0 (1 + nt) e^{-nt} \quad (1.27)$$

Движение груза является аperiодическим. При $t \rightarrow \infty$ выражение (1.27) оказывается неопределенностью типа $0 \cdot \infty$. Для раскрытия неопре-

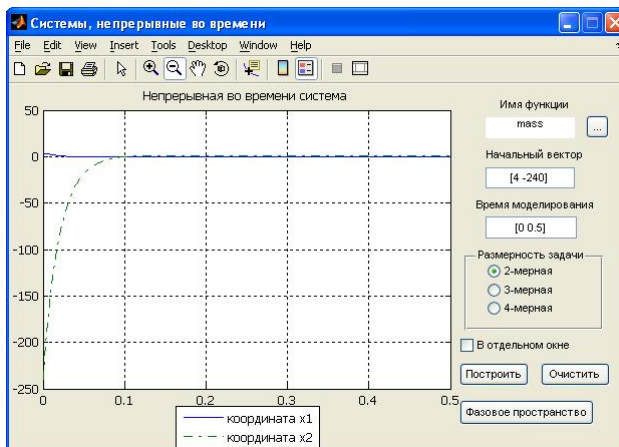


Рис. 1.9: Затухающие колебания груза в жидкости при $n > k$

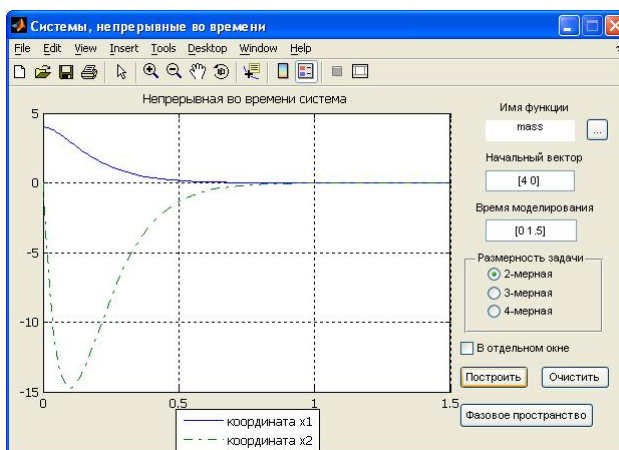


Рис. 1.10: Затухающие колебания груза в жидкости при $n = k$

деленности применяем правило Лопиталья, предварительно представив (1.27) в виде

$$x(t) = \frac{x_0(1 + nt)}{e^{nt}}$$

Тогда

$$\lim_{t \rightarrow \infty} x = \lim_{t \rightarrow \infty} \frac{x_0(1 + nt)}{e^{nt}} = \lim_{t \rightarrow \infty} \frac{\frac{d}{dt}[x_0(1 + nt)]}{\frac{d}{dt}[e^{nt}]} = \lim_{t \rightarrow \infty} \frac{nx_0}{ne^{nt}} = 0$$

Таким образом, при $t \rightarrow \infty$ $x \rightarrow 0$. Итак, груз, согласно уравнению (1.27), совершает аperiodическое затухающее движение. После подстановки численных значений $P = 98$ Н, $c=10$ Н/см, $\gamma = 2$ нсек/см, $x_0 = 4$ см, $v_0 = 0$ см/сек, получим $k = 10$ сек $^{-1}$, $n = 10$ сек $^{-1}$ (см. рис. 1.10)

и

$$x(t) = 4e^{-10t}(1 + 10t)$$

Во всех вышерассмотренных случаях груз совершал свободные колебания при наличии силы сопротивления, пропорциональной скорости. Рассмотрим случай вынужденных колебаний.

Пусть возмущающая сила изменяется по гармоническому закону

$$S = H \sin(pt + \delta)$$

Дифференциальное уравнение вынужденных колебаний материальной точки при наличии силы сопротивления, пропорциональной скорости, имеет вид

$$\ddot{x} + 2n\dot{x} + k^2x = h \sin(pt + \delta)$$

где

$$k^2 = \frac{c}{m} \quad n = \frac{\gamma}{2m} \quad h = \frac{H}{m}$$

Тогда уравнение движения материальной точки будет: $x = x_1 + x_2$, где x_1 — общее решение соответствующего однородного дифференциального уравнения; x_2 — частное решение неоднородного дифференциального уравнения.

Решение x_1 в зависимости от соотношений $n < k$, $n > k$, $n = k$ получается в одном из рассмотренных выше видов (случай 1, случай 2, случай 3).

Частное решение x_2 , определяющее вынужденные колебания, равно

$$x_2 = a \sin(pt + \delta - \varepsilon),$$

где

$$a = \frac{h}{\sqrt{(k^2 - p^2)^2 + 4n^2p^2}}$$
$$\varepsilon = \arctg\left(\frac{2np}{k^2 - p^2}\right)$$

Введя для сокращения записи обозначение

$$A(t) = \frac{h}{\sqrt{(k^2 - p^2)^2 + 4n^2p^2}} \sin(pt + \delta - \varepsilon)$$

получим уравнение движения материальной точки в следующем окончательном виде:

$$x = \begin{cases} e^{-nt} (C_1 \sin(\sqrt{k^2 - n^2}t) + C_2 \cos(\sqrt{k^2 - n^2}t)) + A(t) & \text{при } n < k \\ e^{-nt} (C_1 + C_2 t) + A(t) & \text{при } n = k \\ e^{-nt} (C_1 e^{\sqrt{n^2 - k^2}t} + C_2 e^{-\sqrt{n^2 - k^2}t}) + A(t) & \text{при } n > k \end{cases}$$

Постоянные интегрирования C_1 и C_2 определяются по заданным начальным условиям движения (при $t = 0$ $x = x_0, \dot{x} = \dot{x}_0$).

Так как во всех трех случаях ($n < k$, $n = k$ и $n > k$) в выражение x_1 входит e^{-nt} , то движения x_1 быстро затухают, в то время как амплитуда вынужденных колебаний a , круговая частота вынужденных колебаний и сдвиг между фазами ε от времени не зависят. Круговая частота p вынужденных колебаний x_2 равна круговой частоте возмущающей силы S . Это означает, что сила сопротивления движению, пропорциональная первой степени скорости, не влияет на величину круговой частоты вынужденных колебаний материальной точки.

1.2.2 Модели фракталов

«**Фракталом** называется структура, состоящая из частей, в каком-то смысле подобных целому» [15].

Фракталы окружают нас повсюду. Изрезанные береговые линии, изломанные поверхности горных хребтов, причудливые очертания облаков, раскидистые ветви деревьев, разветвленные сети кровеносных сосудов и нейронов, вспененные потоки бурных рек - все это фракталы. Одни фракталы, типа облаков и горных потоков, постоянно изменяются, другие, подобные деревьям и нейронным сетям, сохраняют свою форму неизменной [13].

Широчайшее распространение фрактальных структур объясняется их *разномасштабностью и самоподобием*: и большие, и малые масшта-

бы фрактальных структур имеют одинаковый закон построения. Форма фрактальной структуры, разглядываемая в микроскоп с любым увеличением, видится одной и той же. Это геометрическое подобие и есть основной принцип роста всего живого, который называют также иерархическим принципом организации (законы ветвления самой тонкой веточки дерева абсолютно те же, что и для всех его ветвей, и для всего ствола в целом).

Задать фрактальную структуру - значит задать не застывшую, неизменную форму, а принцип роста, закон изменения формы. Как правило, алгоритм построения формы гораздо проще, чем полученная с его помощью форма. Фрактал дает компактный способ описания самых замысловатых форм. «Фрактал не есть конечная форма (фрактал никто никогда не видел, так же как число π), а есть закон построения этой формы. Фрактал аккумулирует в себе идею роста» [7].

Осознание этой идеи привело к тому, что понятие фрактала стало широко использоваться в научных исследованиях, и было обнаружено большое число задач, в которых фрактальная структура и размерность служат основными характеристиками системы. Например, в турбулентности теория фракталов теснейшим образом связана с теорией масштабной инвариантности Колмогорова. Скорость турбулентного потока (как функция пространственных переменных и времени) — фрактал, аналогичный броуновской кривой, только с иными локальными свойствами [24].

Фракталы, получаемые с помощью электронных вычислительных машин, принято делить на три группы: **алгебраические**, **геометрические** и **стохастические**. Первые две группы являются детерминированными фракталами, а последняя группа относится к недетерминированным.

История фракталов началась с *геометрических* фракталов, которые исследовались математиками в XIX веке [16]. Фракталы этого класса — самые наглядные, потому что в них сразу видна самоподобность. При-

мерами таких фракталов являются треугольник Серпинского, снежинка Коха, кривая Леви и многие другие.

Алгебраические фракталы являются самой большой группой фрактал: и получаются с помощью нелинейных процессов в n -мерных пространствах. Самыми известными из них являются множества Мандельброта и Жюлиа, бассейны Ньютона и т.д.

Стохастические фракталы получаются при внесении в итерационный процесс помех с желаемыми стохастическими параметрами. Эта группа фракталов применяется при моделировании рельефа местности, поверхности моря и формы облаков [28].

Глава 2

Компьютерная реализация математических моделей

Компьютерная модель — это программная реализация математической модели, дополненная различными служебными программами (например, рисующими и изменяющими графические образы во времени). Компьютерная модель имеет две составляющие — программную и аппаратную. Программная составляющая так же является абстрактной знаковой моделью. Это лишь другая форма абстрактной модели, которая, однако, может интерпретироваться уже не только математиками и программистами, но и техническим устройством — процессором компьютера [9].

Суть компьютерного моделирования заключена в получении количественных и качественных результатов по имеющейся модели. Качественные выводы, получаемые по результатам анализа, позволяют обнаружить неизвестные ранее свойства сложной системы: ее структуру, динамику развития, устойчивость, целостность и др. Количественные выводы, в основном, носят характер прогноза некоторых будущих или объяснения прошлых значений переменных, характеризующих систему. Компьютерное моделирование для порождения новой информации использует любую информацию, которую можно актуализировать с помощью ЭВМ.

Основные функции компьютера при моделировании:

- выполнять роль вспомогательного средства для решения задач, решаемых обычными вычислительными средствами, алгоритмами, технологиями;
- выполнять роль средства постановки и решения новых задач, не решаемых традиционными средствами, алгоритмами, технологиями;
- выполнять роль средства конструирования компьютерных обучающе - моделирующих сред;
- выполнять роль средства моделирования для получения новых знаний;
- выполнять роль «обучения» новых моделей (самообучающиеся модели).

2.1 Среда разработки средства электронной поддержки курса «Основы моделирования»

На исторически ранних этапах компьютерного моделирования эти программы создавались либо на языках программирования высокого уровня (Фортране, Алголе), либо даже на языке низкого уровня Ассемблере. Применявшиеся в те годы технологии программирования требовали на создание моделей очень много времени.

Начиная с 1960-х гг. было разработано множество систем моделирования или пакетов математических вычислений самого разного уровня — от рассчитанной «на всех» системы Mathcad, поразительно компактной, быстрой и удобной для простых символьных вычислений системы Derive и до компьютерных монстров Mathematica, Matlab и Maple, имеющих тысячи встроенных и библиотечных функций и изумительные возможности графической визуализации вычислений.

В системе Matlab используется ядро системы Maple, придавая ей необычные для нее возможности символьной математики.

Вряд ли мощная математическая система Matlab, разделяющая претензии на мировое лидерство с системами класса Mathematica, нужна секретарше или даже директору небольшой коммерческой фирмы. Но, несомненно, любая серьезная научная лаборатория или кафедра вуза должны располагать подобной системой, если они всерьез заинтересованы в автоматизации выполнения математических расчетов любой степени сложности. Несмотря на свою направленность на серьезные математические вычисления, системы класса Matlab необходимы довольно широкой категории пользователей: студентам и преподавателям вузов, инженерам, аспирантам, научным работникам и даже учащимся математических классов общеобразовательных и специальных школ. Все они найдут в Matlab многочисленные достойные возможности для применения.

Сравнение системы Matlab с лидером среди систем компьютерной математики - системой Mathematica - непродуктивно. У каждой программы есть свои достоинства и недостатки. А главное - у них есть свои приверженцы, которых бесполезно убеждать, что иная система в чем-то лучше. Все, кто всерьез применяют системы компьютерной математики, должны работать с несколькими системами, ибо только это гарантирует высокий уровень надежности сложных вычислений.

И все же надо отметить, что интерфейс Matlab более интуитивно понятен, чем у строгой Mathematica. Matlab имеет мощную графику, простота управления параметрами и легкость подготовки графических процедур часто позволяет визуализировать решения математических задач с меньшими усилиями, чем при использовании системы Mathematica. Обе системы в последних реализациях сделали качественный скачок в направлении эффективности решения задач в численном виде, в частности за счет повышения скорости выполнения матричных операций.

Особенно эффективно использование Matlab при обучении математике [1]. Высочайший «интеллект» этой системы символьной математики объединяется в ней с прекрасными средствами математического числен-

ного моделирования и просто потрясающими возможностями графической визуализации решений. Применение таких систем, как Matlab, возможно при преподавании и самообразовании от самых основ до вершин математики.

Учитывая начальную подготовку студентов в области программирования и опираясь на вышеуказанные объективные причины, в качестве среды разработки средства электронной поддержки курса «Основы моделирования» выбрана система Matlab.

2.2 Назначение, структура и возможности программы

В рамках дипломной работы была написана программа BaseOfModelling (Листинг программы, Приложение А). Она разрабатывалась как средство электронной поддержки курса «Основы моделирования».

Спектр задач, исследование которых может быть осуществлено при помощи программы BaseOfModelling, охватывает:

1. Программы, для иллюстрации как непрерывных, так и дискретных во времени систем, как линейных, так и нелинейных вида $\dot{x} = f(x)$ или $x(k+1) = f(x(k))$, размерностью ≤ 4 ;
2. Программы для моделирования так называемых градиентных систем, т.е. систем вида $\dot{x} = -\nabla f(x)$, где

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

3. Набор программ для рисования комплексных динамических систем и фракталов.

В состав программы BaseOfModelling входят приложения с графическим интерфейсом пользователя, которые обеспечивают быстрый и наглядный доступ к основным функциям.

Удобная справочная система содержит подробные руководства пользователя, описание основных функций и достаточно большое число примеров их использования.

Программа BaseOfModelling выводит, по окончании вычислений, траектории движения координат, фазовое пространство, позволяет сохранять полученные графики или распечатывать их для отчета.

Все данные, полученные в ходе программы могут быть проверены путем сравнения графиков численного и аналитического решений.

Пользователь — студент с незначительной (начальной) подготовкой в области программирования - должен выполнять вручную следующие действия:

1. писать m-файлы для функции $f(x)$;
2. запускать моделирующую программу;
3. вводить данные;
4. обрабатывать полученные данные;
5. сохранять графики решения.

Моделирование динамических систем, непрерывных во времени

Для решения систем вида $\dot{x} = f(x)$ использовался Метод Рунге-Кутты 4 порядка [11], т. е. на каждом шаге вычислялись

$$\begin{aligned}\eta_1^i &= f(t_i, x_i) \\ \eta_2^i &= f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}\eta_1^i\right) \\ \eta_3^i &= f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}\eta_2^i\right) \\ \eta_4^i &= f(t_i + h, x_i + h\eta_3^i) \\ \Delta x_i &= \frac{h}{6} (\eta_1^i + 2\eta_2^i + 2\eta_3^i + \eta_4^i) \\ x_{i+1} &= x_i + \Delta x_i\end{aligned}$$

где $t \in [t_0, t_1]$ — время моделирования, $x(t_0) = x_0$ — начальный вектор, $h = \frac{t_1 - t_0}{n}$ — расчетный шаг, n — число шагов, t_0 — начальный момент моделирования, t_1 — конечный момент моделирования, Δx_i — средневзвешенная величина поправок $h\eta_1^i, h\eta_2^i, h\eta_3^i, h\eta_4^i$ каждого этапа с весовыми коэффициентами $\frac{1}{6}, \frac{2}{6}, \frac{2}{6}, \frac{1}{6}$.

Используя данный метод, в ходе работы программы было получено решение системы Лоренца (рис. 2.1)

$$\dot{x}_1 = \sigma(x_2 - x_1)$$

$$\dot{x}_2 = rx_1 - x_2 - x_1x_3$$

$$\dot{x}_3 = x_1x_2 - bx_3$$

где $\sigma = 10, b = \frac{8}{3}, r = 28$, и диаграмма фазового пространства (рис. 2.2).

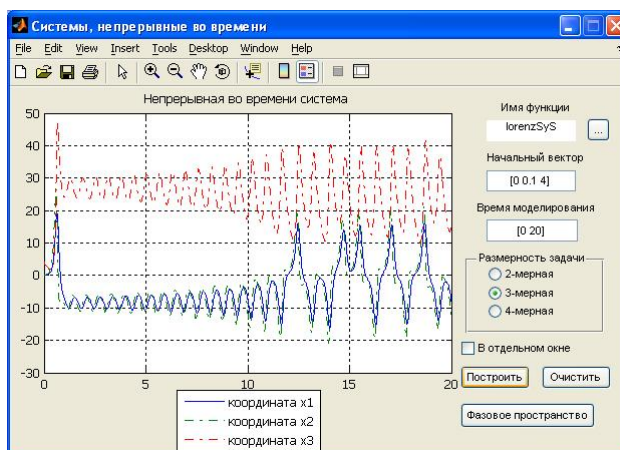


Рис. 2.1: Зависимость координат x_1, x_2, x_3 от времени t

Моделирование дискретных динамических систем

Для решения систем вида $x(k+1) = f(x(k))$ использовался Метод итерации, т. е.

$$x(k+1) = f(k, x(k))$$

где $k \in [t_0, t_1]$, t_0 — начальный момент моделирования, t_1 — конечный момент моделирования.

Используя данный метод, в ходе работы программы было получено решение системы (1.10) (см. рис. 1.5, 1.6).

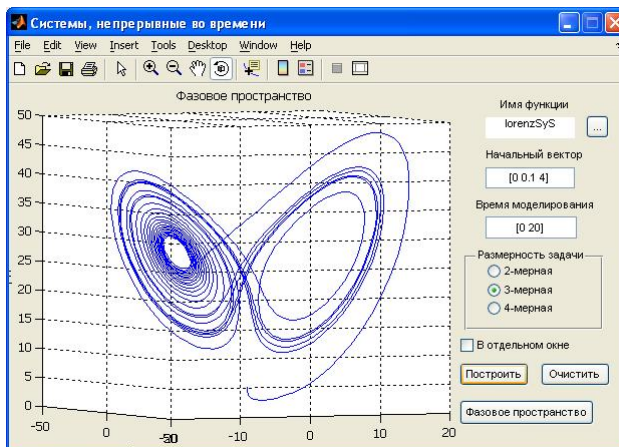


Рис. 2.2: Фазовая траектория

Моделирование градиентных динамических систем

Для решения систем вида $\dot{x} = -\nabla f(x)$, где

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

использовался алгоритм, аналогичный моделированию динамических систем, непрерывных во времени (вместо $f(x)$ использовался $-\nabla f(x)$).

Используя данный метод, в ходе работы программы было получено решение системы (рис. 2.3)

$$\dot{x}_1 = \cos x_1 \sin x_2 \quad (2.1)$$

$$\dot{x}_2 = \cos x_2 \sin x_1 \quad (2.2)$$

Построение геометрических фракталов с помощью систем итерируемых функций

Теория систем итерируемых функций является одним из наиболее замечательных и глубоких достижений теории фракталов. Подход на основе систем итерируемых функций представляет хорошую теоретическую базу для математического исследования многих классических фракталов и их обобщений. Теория итерированных функций является составной частью общей теории динамических систем.

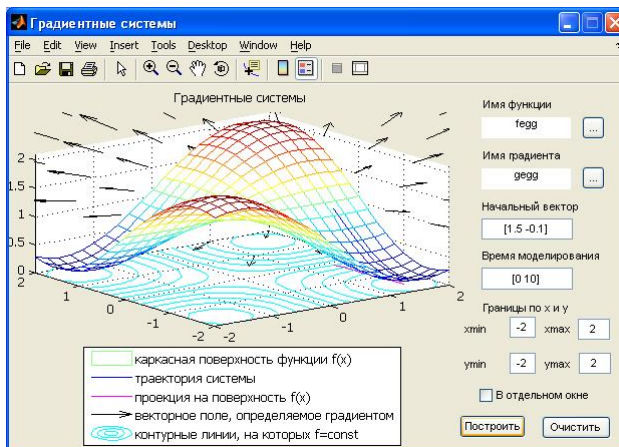


Рис. 2.3: Градиентная система (2.1)-(2.2)

Системой итерируемых функций будем называть конечное множество сжимающих отображений $F = f_1 \cup f_2 \cup \dots \cup f_k$, где

$$f_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

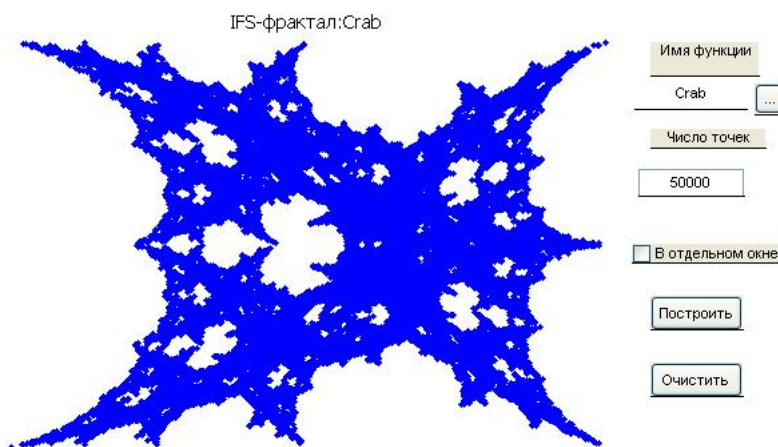


Рис. 2.4: Фрактал, полученный с помощью системы итерируемых функций A_1

Идея алгоритма построения фрактала на основе систем итерируемых функций весьма проста [23]. Мы начинаем с произвольного компактного набора X , применяем к нему одно из преобразований f_i и выводим на экран.

Экран — фактически множество точек, «включают» те точки, которые являются частью изображения и «выключают» те точки, которые

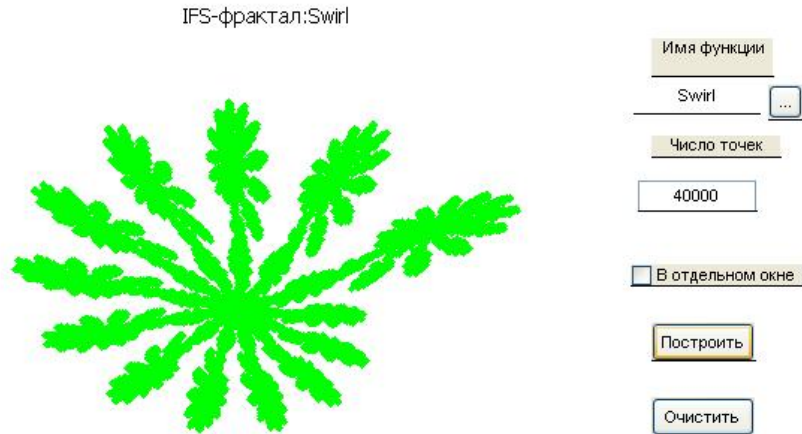


Рис. 2.5: Фрактал, полученный с помощью системы итерируемых функций A_2

— нет. Таким образом, компьютерный экран можно считать матрицей 0 и 1. 0 представляют точки, которые не находятся в заданном множестве, в то время как, 1 представляют точки, которые находятся в нем.

Предполагается, что фрактал находится в единичном квадрате, то есть, все его x -, и y -координаты находятся между 0 и 1.

Шаг 0:

Пусть A — матрица $6 \times k$ — система итерируемых функций. $NPTS$ — число точек на плоскости.

Для всех $i := 0..NPTS + 50$ выполнить следующую процедуру:

Шаг 1:

Положить $X = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ (или любое другое начальное значение).

Шаг 2:

Увеличить счетчик $i := i + 1$, выбрать одно из преобразований j случайным образом из множества от 1 до k .

Шаг 3:

Вычислить $X = f_j(X)$, где f_j — j -ый столбец матрицы A .

Шаг 4:

1. Если $i > 50$, то вывести точку на экран.

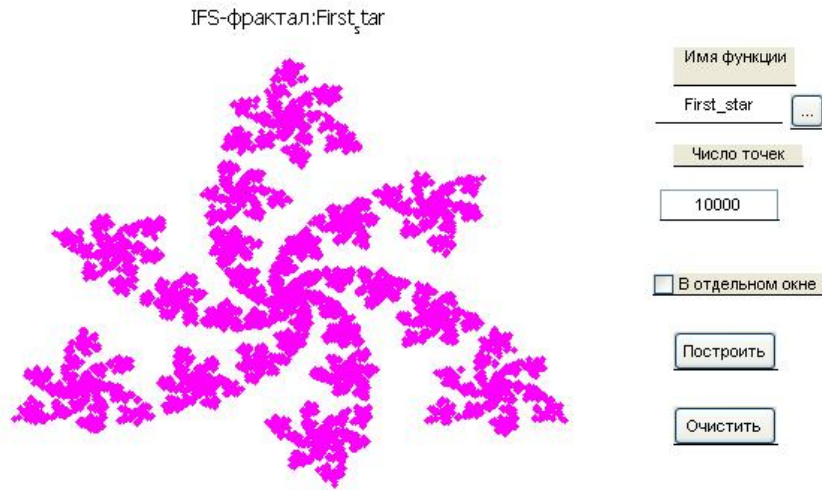


Рис. 2.6: Фрактал, полученный с помощью системы итерируемых функций A_3

2. Если $i < NPTS + 50$, то переход на шаг 2, иначе конец.

Пусть

$$A_1 = \begin{bmatrix} 0.5 & 0.2 & 0 & 0.5 & 0 & 0 \\ -0.5 & 0 & 0.2 & 0.5 & 1 & 0 \\ 0.5 & 0.2 & 0 & -0.5 & 0 & 1 \\ -0.5 & 0 & -0.2 & -0.5 & 1 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0.745455 & -0.459091 & 0.406061 & 0.887121 & 1.460279 & 0.691072 \\ -0.424242 & -0.065152 & -0.175758 & -0.218182 & 3.809567 & 6.741476 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.255 & 0 & 0 & 0.255 & 0.006 & 7.307 \\ 0.255 & 0 & 0 & 0.255 & -3.09 & 1.931 \\ 0.255 & 0 & 0 & 0.255 & 3.102 & 1.931 \\ 0.268485 & -0.737658 & 0.737658 & 0.268485 & 3.705256 & 3.658438 \end{bmatrix}$$

Используя данный алгоритм, в ходе работы программы получены изображения для A_1 , A_2 , A_3 (соответственно рис. 2.4, 2.5, 2.6).

Построение комплексных (алгебраических) фракталов

Для построения алгебраических фракталов используются итерации нелинейных отображений, задаваемых простыми алгебраическими формулами.

Наиболее изучен двухмерный случай [17]. Нелинейные динамические системы могут обладать несколькими устойчивыми состояниями. Каждое устойчивое состояние (аттрактор) обладает некоторой областью начальных состояний, при которых система обязательно в него перейдет. Таким образом, фазовое пространство разбивается на области притяжения аттракторов.

Если фазовым является двухмерное пространство, то, окрашивая области притяжения различными цветами, можно получить цветовой фазовый портрет этой системы (итерационного процесса). Меняя алгоритм выбора цвета, можно получить сложные фрактальные картины с причудливыми многоцветными узорами.

Алгоритм построения достаточно прост и основан на итеративном выражении: $z_{i+1} = F(z_i)$, где $F(z)$ - какая-либо функция комплексной переменной.

Области притяжения и множества Жюлиа

Для построения множества Жюлиа используется $F(z) = z^2 + c$, где $c = const$.

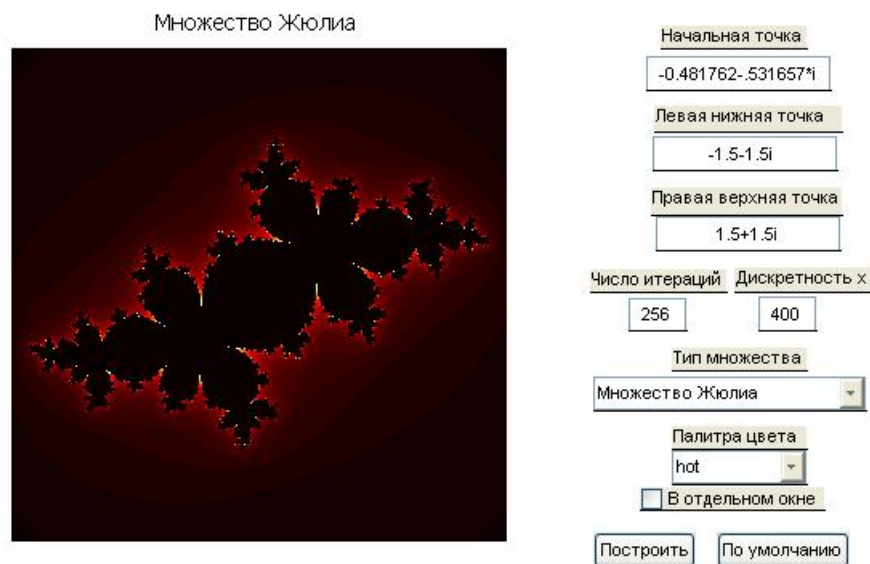


Рис. 2.7: Множество Жюлиа: $c = -0.481762 - 0.531657i$

Запишем комплексные числа z и c в виде $z = x + iy$, $c = p + iq$.

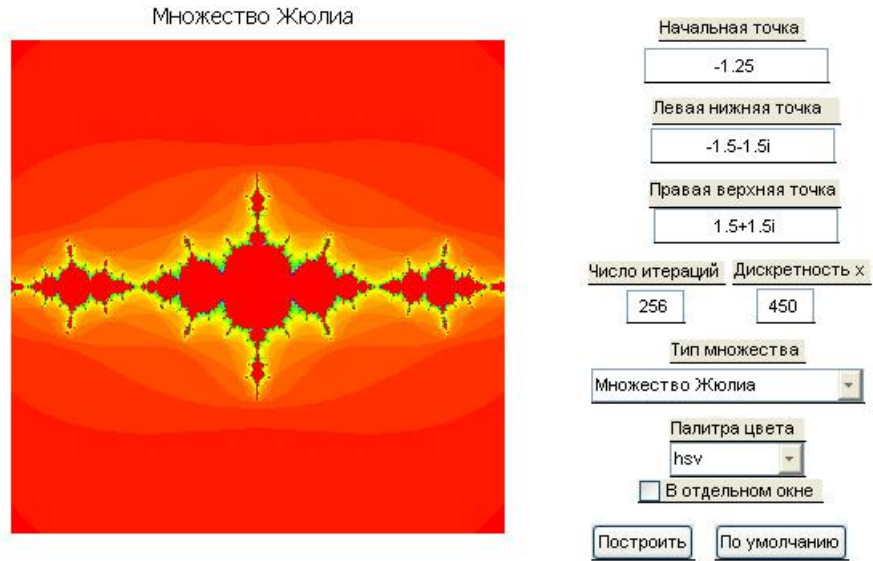


Рис. 2.8: Множество Жюлиа: $c = -1.25$

Тогда закон, которым описывается процесс, примет вид:

$$\begin{aligned} x_{k+1} &= x_k^2 - y_k^2 + p \\ y_{k+1} &= 2x_k y_k + q \end{aligned} \quad (2.3)$$

Поскольку бесконечность — всегда аттрактор для такого процесса, зададимся целью раскрасить ее область притяжения. Установим соответствие между цветом и временем, за которое точка (x, y) уходит на бесконечность. Предположим, что разрешающая способность экрана равна $a \times b$ точек. Предположим также, что он может быть одновременно раскрашен в $K + 1$ цветов, которые мы пронумеруем от 0 до K , причем числу 0 сопоставим черный цвет.

Шаг 0:

Выбрать параметр $c = p + iq$. Выбрать $x_{min} = y_{min}$, $x_{max} = y_{max}$. Положить $\Delta x = \frac{x_{max} - x_{min}}{a-1}$, $\Delta y = \frac{y_{max} - y_{min}}{b-1}$. Для всех пар (n_x, n_y) , где $n_x = 0, \dots, a-1$ и $n_y = 0, \dots, b-1$, выполнить следующую процедуру:

Шаг 1:

Положить $x_0 = x_{min} + n_x \Delta x$, $y_0 = y_{min} + n_y \Delta y$, $k = 0$.

Шаг 2 (итерация):

Вычислить (x_{k+1}, y_{k+1}) по (x_k, y_k) , используя закон (2.3). Увеличить

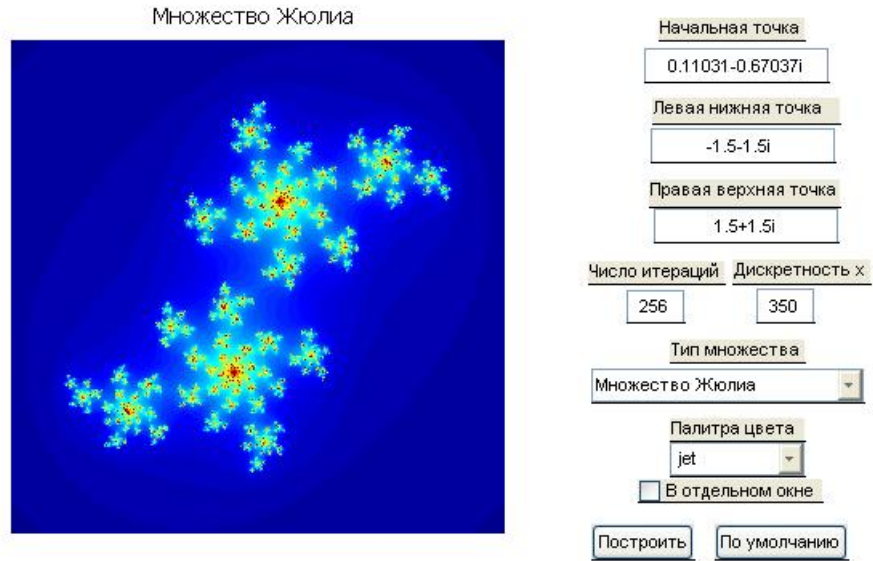


Рис. 2.9: Множество Жюлия: $c = 0.11031 - 0.67037i$

счетчик k на 1 ($k := k + 1$).

Шаг 3 (оценка):

Вычислить $r = x^2 + y^2$.

1. Если $r > 2$, то выбрать цвет k и идти на шаг 4.
2. Если $k = K$, то выбрать цвет 0 (черный) и идти на шаг 4.
3. Если $r \leq 2$, $k < K$, то вернуться на шаг 2.

Шаг 4:

Приписать цвет k точке экрана (n_x, n_y) и перейти к следующей точке, начиная с шага 1.

Используя данный алгоритм, в ходе работы программы получены множества Жюлия при различных начальных значениях (рис. 2.7, 2.8 и 2.9).

ЗАМЕЧАНИЕ: Использование большого числа цветов K (например, $K = 200$) оправданно. Если же количество различных цветов невелико, удобно использовать их периодическим образом. И даже с помощью двух цветов (белого и черного) можно этим способом получить интересные картины. Время вычислений можно уменьшить в 2 раза ввиду симметрии процесса (2.3). Точки (x, y) и $(-x, -y)$ имеют одну и ту же судьбу,

следовательно, картина должна быть симметричной относительно начала координат. Варьируя исходные данные x_{min}, x_{max}, \dots на шаге 0, можно получить сильно увеличенные изображения.

Множество Мандельброта

Для построения множества Мандельброта используется $F(z) = z^2 + c$.

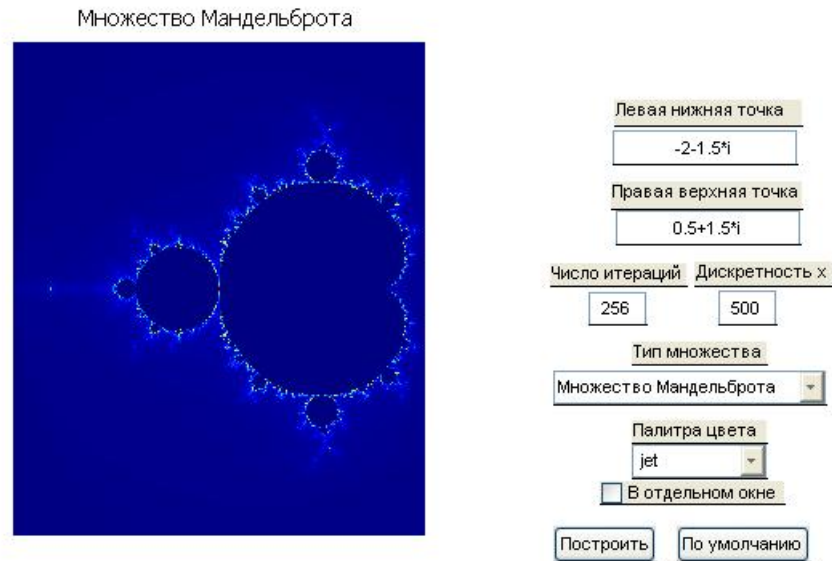


Рис. 2.10: Множество Мандельброта: $-1.5 < Re(c) < 0.5, -1.5 < Im(c) < 1.5$

У процесса (2.3) нет второго (помимо бесконечности) аттрактора, если критическая точка $z_0 = 0$ стремится к бесконечности (z_0 является критической точкой процесса $z \mapsto f(z)$, если $f''(z_0) = 0$). На этом строится следующая процедура, с помощью которой мы получим окрашенное в черный цвет множество Мандельброта с окружением, раскрашенным в разные цвета. Вновь предположим, что экран имеет $a \times b$ точек и $K + 1$ цветов.

Шаг 0:

Выбрать $p_{min}, p_{max}, q_{min}, q_{max}$. Положить $\Delta p = \frac{p_{max} - p_{min}}{a - 1}, \Delta q = \frac{q_{max} - q_{min}}{b - 1}$. Для всех точек (n_p, n_q) экрана $n_p = 0, \dots, a - 1, n_q = 0, \dots, b - 1$ выполнить следующую процедуру:

Шаг 1:

Положить $p_0 = p_{min} + n_p \Delta p, q_0 = q_{min} + n_q \Delta q, k = 0, x_0 = y_0 = 0$.

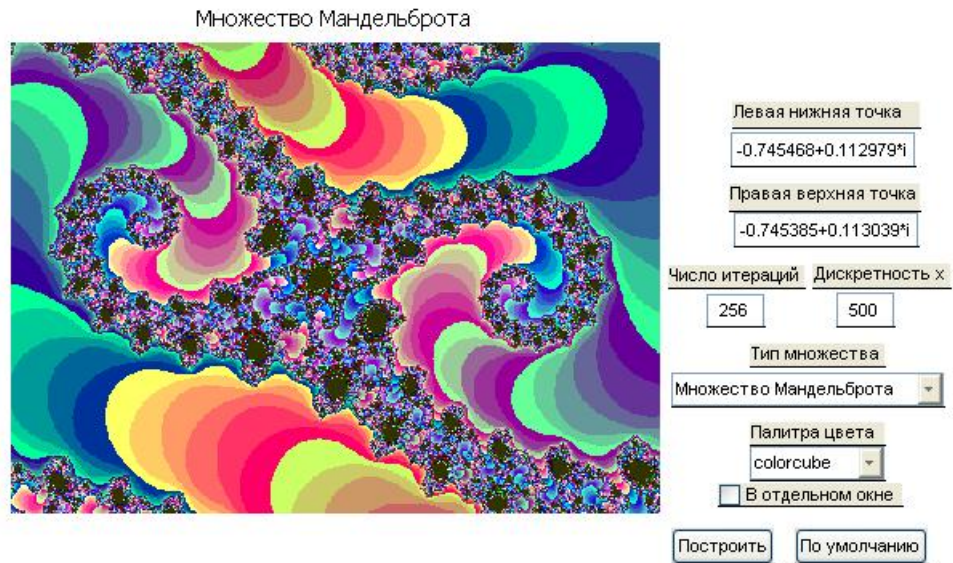


Рис. 2.11: Множество Мандельброта: $-0.745468 < Re(c) < -0.745385, 0.112979 < Im(c) < 0.113039$

Шаг 2:

Вычислить (x_{k+1}, y_{k+1}) с помощью (2.3). Затем положить $k := k + 1$.

Шаг 3:

Вычислить $r = x^2 + y^2$.

1. Если $r > 2$, то выбрать цвет k и идти на шаг 4.
2. Если $k = K$, то выбрать цвет 0 (черный) и идти на шаг 4.
3. Если $r \leq 2, k < K$, то вернуться на шаг 2.

Шаг 4:

Приписать цвет к точке экрана (n_p, n_q) и перейти к следующей точке (шаг 1).

Используя данный алгоритм, в ходе работы программы получены множество Мандельброта (рис. 2.10) и крупные планы (рис. 2.11 и 2.12).

Бассейны Ньютона

Области с фрактальными границами появляются при приближенном нахождении корней нелинейного уравнения $f(z) = 0$ алгоритмом Ньютона на комплексной плоскости. Оказывается, что на границе между

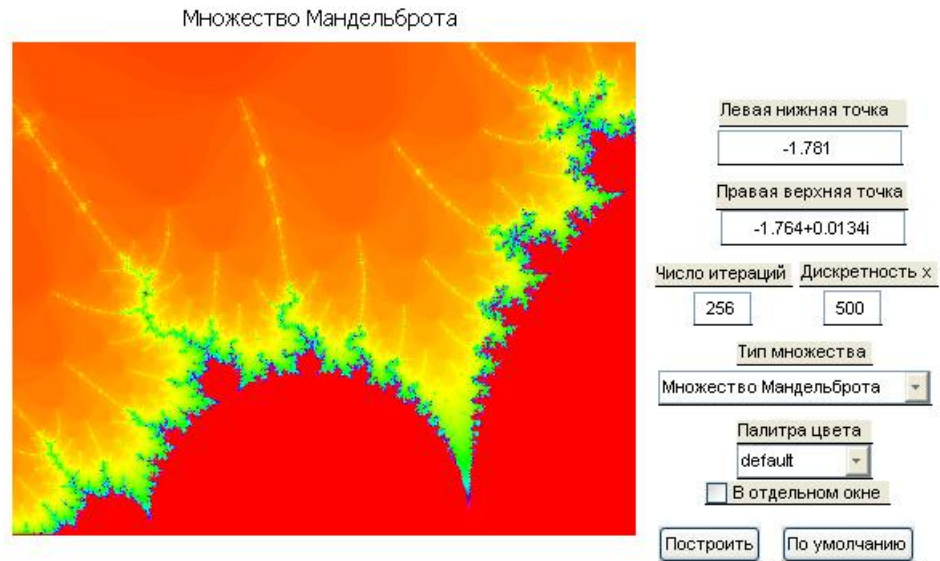


Рис. 2.12: Множество Мандельброта: $-1.781 < \text{Re}(c) < -1.764, 0 < \text{Im}(c) < 0.0134$

любыми двумя цветами всегда расположена гирлянда островков третьего цвета. Границы этих островков, в свою очередь, состоят из гирлянд островков меньшего размера соответствующего дополнительного цвета и т. д. Непрерывно уменьшаясь в размерах, детали границы постоянно воспроизводят самих себя. В результате оказывается, что каждая точка такой фрактальной границы соседствует сразу с тремя областями притяжения.

Рассмотрим простейшее уравнение третьей степени $z^3 - 1 = 0$. В этом случае имеем три кубических корня из 1, а именно, $\omega_1 = 1$, $\omega_2 = -1/2 + i$ и $\omega_3 = -1/2 - i$. Можно легко показать, что эти корни действительно являются притягивающими неподвижными точками отображения

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \text{ с } f(z) = z^3 - 1, \text{ т.е.}$$

$$z_{n+1} = z_n - (z_n^3 - 1)/3z_n^2.$$

Полагая $z = x + yi$ и разделяя вещественную и мнимую части,

приходим к двумерному вещественному отображению:

$$x_{k+1} = \frac{2}{3}x_n + \frac{x_n^2 - y_n^2}{3(x_n^2 + y_n^2)^2}$$
$$y_{k+1} = \frac{2}{3}x_n \left[1 - \frac{x_n}{(x_n^2 + y_n^2)^2} \right]$$

Используя алгоритм построения множества Жюлиа (при $c = 0$), в ходе работы программы получен фрактал (бассейн) Ньютона (рис. 2.13).

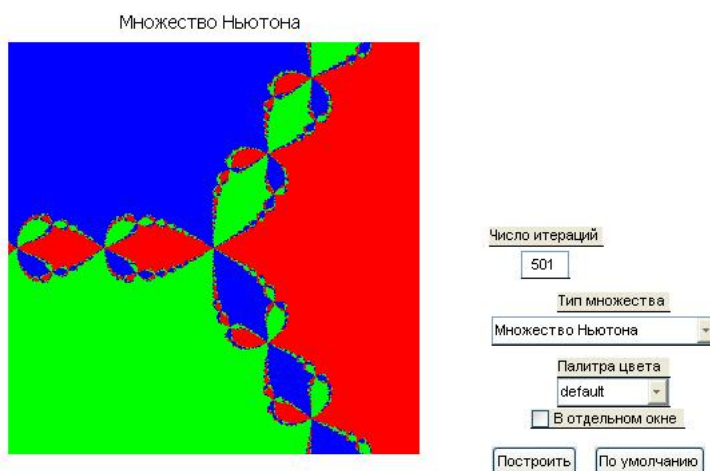


Рис. 2.13: Множество Ньютона для $f(z) = z^3 - 1$

Оценка адекватности компьютерной реализации математических моделей

Сопоставление полученных в результате работы программы данных с результатами качественного анализа (аналитического решения), анализа натурального эксперимента и данными, полученными с помощью других численных алгоритмов, доказывает (показывает) непротиворечивость, правильность вычислений.

Заключение

В данной работе на основе построенных математических моделей реальных систем **разработана программа BaseOfModelling**. Она является программно-методической поддержкой курса «Основы моделирования», содержит в себе подробное руководство пользователя и набор задач, рекомендуемых к моделированию. Программа BaseOfModelling может быть внедрена в процесс обучения в 2007/2008 учебном году на факультете Математики и информационных технологий в первом семестре в рамках практических занятий курса «Основы моделирования».

Изучив литературу по вопросам математического моделирования, можно сделать вывод о том, что в большинстве источников для моделирования применяются только специальные пакеты, требующие глубоких знаний как в программировании, так и в рассматриваемой области, и не охватывается начальный уровень подготовки.

Приведенные примеры и разработанная программа рассчитаны на студентов математических специальностей с начальной подготовкой в области программирования и владения математическим аппаратом. Представляется целесообразным начинать работу в математическом моделировании именно с простейших моделей, не отягченных математическими сложностями. Будущий специалист, избравший этот не самый легкий путь, должен обладать широкими и глубокими знаниями не только во многих разделах «чистой» и «прикладной» математики и информатики, но и аналогичными познаниями в относящихся к объекту моделирования естественно-научных (физике, механике, химии и др.) и/или гуманитарных дисциплинах. Только доскональное знание объекта моделирования,

соответствующей предметной области, а также возможность говорить на одном языке со специалистами-«заказчиками» модели позволяет надеяться на успешную реализацию того или иного проекта по созданию математической модели процесса или явления. По крайней мере, такими качествами должен обладать «постановщик» задачи построения той или иной модели.

И в заключении нами выдвигается предположение о том, что *применение* разработанной программы *позволит лучше освоить курс* «Основы моделирования», пробудит интерес к математике в целом, и к моделированию в частности.

Библиографический список

- [1] Ануфриев И. Е., Смирнов А. Б., Смирнова Е. Н. Matlab 7. - СПб.: БХВ-Петербург, 2005. 1083 с.
- [2] Бать М. И. Теоретическая механика в примерах и задачах. Т 2. - М.: Наука, 1972. 623 с.
- [3] Бенькович Е.С. Практическое моделирование. - М.: Наука, 1999. 365 с.
- [4] Бондаренко В.А., Дольников В.Л. Фрактальное сжатие изображений по Барнсли-Слоану. - М.: Мир, 1978. 106 с.
- [5] Введение в математическое моделирование: Учебное пособие/ Под ред. П. В. Трусова. - М.: Логос, 2005. 440 с.
- [6] Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения): Учеб. пособие для вузов. - М.: ГУП «Издательство «Высшая школа», 2001. 382 с.
- [7] Волошинов А.В. Математика и искусство. - М.: Просвещение, 2002. 399 с.
- [8] Горстко А. Б. Познакомьтесь с математическим моделированием. - М.: Знание, 1991. 156 с.
- [9] Гулд Х., Тобочник Я. Компьютерное моделирование в физике. Ч.2. - М.: Мир, 1990. 320 с.
- [10] Дьяконов В.П. Matlab 6/6.1/6.5+Simulink 4/5. Основы программирования: Руководство пользователя. - М.: Солон-Пресс, 2002. 768 с.

- [11] Дьяконов В.П. Matlab. Анализ, идентификация и моделирование систем : Специальный справочник / Дьяконов Владимир, В. Круглов. - СПб. : Питер, 2002. - 448 с.
- [12] Кривилев А. Основы компьютерной математики с использованием системы Matlab. - М.: Лекс-книга, 2005. 436 с.
- [13] Кроновер Р.М. Фракталы и хаос в динамических системах. - М.: Постмарист, 2000. 352 с.
- [14] Лазарев Ю. Моделирование процессов и систем в MATLAB: учеб. курс / Лазарев Юрий. - СПб.: Питер ВHV, 2005. 512 с.
- [15] Мандельброт Б. Фрактальная геометрия природы. - М.: Институт компьютерных исследований, 2002. 656 с.
- [16] Морозов А.Д. Введение в теорию фракталов. - М.: Институт компьютерных исследований, 2002. 160 с.
- [17] Пайтген Х.-О., Рихтер П.Х. Красота фракталов. - М.: Мир, 1993. 295 с.
- [18] Рыжиков Ю.И. Имитационное моделирование. М.: Логос, 2003. 357 с.
- [19] Стивенс Р. Delphi. Готовые алгоритмы. - СПб.: Питер, 2004. 508 с.
- [20] Самарский А.А. Математическое моделирование. М.: ФИЗМАТЛИТ, 2002. 347 с.
- [21] Советов Б.Я. Моделирование систем: Практикум. - М.: Высшая школа, 2003. 295 с.
- [22] Советов Б. Я. Моделирование систем : Учебник для вузов. М.: Высшая школа, 2001. 343 с.
- [23] Федер Е. Фракталы. - М.: Мир, 1991. 284 с.
- [24] Фоменко А. Т. Наглядная геометрия и топология. - М.: Изд-во МГУ, 1993. 302 с.

- [25] Фракталы в физике. Труды 6-го международного симпозиума по фракталам в физике. - М.: Мир, 1988. 247 с.
- [26] Чуличков А.И. Математические модели нелинейной динамики. - СПб.: Питер, 2002. 350 с.
- [27] Шеннон Р. Имитационное моделирование систем — искусство и наука. - М.: Мир, 1978. 154 с.
- [28] Шредер М. Фракталы, хаос, степенные законы. Миниатюры из бесконечного рая. - Ижевск: РХД, 2001. 203 с.
- [29] Edward R. Scheinerman Invitation to dynamical system. - Prentice-Hall, 1995. 220 с.

Приложение А

Листинг программы BaseOfModelling

Моделирование динамических систем, непрерывных во времени

```
function varargout = contSyS(varargin)
global RHSNAME;%имя функции
RHSNAME=get(handles.EdtF,'string'); global x;
ParStr = ['x0=', get(handles.EdtV,'String'), ','];%начальный вектор
% Выполнение строки, запись начального приближения в вектор x0
eval(ParStr); [r,c] = size(x0); if (r==1), x0 = x0'; end;
%начальный и конечный момент моделирования
ParStr = ['t0=', get(handles.EdtB,'String'), ','];
% Выполнение строки, запись начального приближения в вектор t0
eval(ParStr); options = odeset('RelTol',1e-4);
%решение дифференциального уравнения методом Рунге-Кутты
[t,x] = ode45(@RHSpatch, t0 ,x0,options); global flag1;
%в отдельном ли окне?
if get(handles.cbInd,'Value'), figure; flag1=1;else flag1=0;end;
if get(handles.rb2,'value') %если задача двумерная
    plot(t,x(:,1),'-',t,x(:,2),'-.')
end
if get(handles.rb3,'value') %если задача трехмерная
    plot(t,x(:,1),'-',t,x(:,2),'-.',t,x(:,3),'-.')
end
```

```

if get(handles.rb4,'value') %если задача четырехмерная
    plot(t,x(:,1),'-',t,x(:,2),'-.',t,x(:,3),'-.',t,x(:,4),'-.')
end title('Непрерывная во времени система') legend('координата 1',
'координата x2','координата x3','координата
x4','Location','SouthOutside');
grid on;%отображение сетки

```

Моделирование дискретных динамических систем

```

function varargout = discrSyS(varargin) function
btnPlot_Callback(hObject, eventdata, handles)
global RHSNAME;%имя функции
RHSNAME=get(handles.EdtF,'string'); global x;
%начальный вектор
ParStr = ['x0=', get(handles.EdtV,'String'), ';''];
% Выполнение строки, запись начального приближения в вектор x0
eval(ParStr); [r,c] = size(x0); if (r==1), x0 = x0'; end;
%начальный и конечный момент моделирования
ParStr = ['t0=', get(handles.EdtB,'String'), ';''];
% Выполнение строки, запись начального приближения в вектор t0
eval(ParStr); k = [t0(1):t0(2)]; n = length(k); x =
zeros(n,length(x0)); x(1,:) = x0';
%решение методом итерации
for j=2:n
    x(j,:) = feval(RHSNAME, x(j-1,:))';
end;
global flag1;%в отдельном ли окне?
if get(handles.cbInd,'Value'), figure; flag1=1;else flag1=0;end;
if get(handles.rb1,'value') %если задача одномерная
    plot(k,x(:,1),'-');
end if get(handles.rb2,'value')
    plot(k,x(:,1),'-',k,x(:,2),'-.')%если задача двумерная
end

```

```

if get(handles.rb3,'value') %если задача трехмерная
    plot(k,x(:,1),'-',k,x(:,2),'-.',k,x(:,3),'--')
end
if get(handles.rb4,'value') %если задача четырехмерная
    plot(k,x(:,1),'-',k,x(:,2),'-.',k,x(:,3),'--',k,x(:,4),':')
end title('Дискретная во времени система'); legend('координата
x1', 'координата x2','координата x3','координата
x4','Location','SouthOutside');

```

Моделирование градиентных динамических систем

```

function varargout = gradSyS(varargin) function
btnPlot_Callback(hObject, eventdata, handles)
global FNAME;%имя функции
FNAME=get(handles.EdtF,'string');
global GNAME; %имя функции градиента
GNAME=get(handles.EdtG,'string') ;
%начальный вектор
ParStr = ['x0=', get(handles.EdtV,'String'), ';''];
% Выполнение строки, запись начального приближения в вектор x0
eval(ParStr);
[r,c] = size(x0);
if (r==1), x0 = x0'; end;
%начальный и конечный момент моделирования
ParStr = ['t0=', get(handles.EdtB,'String'), ';''];
% Выполнение строки, запись начального приближения в вектор t0
eval(ParStr);
%решение дифференциального уравнения методом Рунге-Кутты
options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4]);
[t,x] = ode45('gpatch', t0,x0,options);
xmin = str2double(get(handles.edit6,'String'));
xmax = str2double(get(handles.edit7,'String'));
ymin = str2double(get(handles.edit8,'String'));

```

```

ymax = str2double(get(handles.edit9,'String'));
steps = 20;
xlist = [xmin : (xmax-xmin)/steps : xmax ];
ylist = [ymin : (ymax-ymin)/steps : ymax ];
[xx,yy] = meshgrid(xlist,ylist);
zz = zeros(steps+1,steps+1);
dx = zz;
dy = zz;
for p = 1:steps+1, for q = 1:steps+1
    zz(p,q) = feval(FNAME,[xx(p,q);yy(p,q)]);
    g = feval(GNAME,[xx(p,q);yy(p,q)]);
    dx(p,q) = g(1);
    dy(p,q) = g(2);
end;end;
zmin = min(min(zz));
zmin = min(zmin,0);
zmax = max(max(zz));
zmax = max(zmax,0);
n = length(t);
z = zeros(n,1);
for k=1:n;
    z(k) = feval(FNAME,x(k,:));
end;
if get(handles.cbInd,'Value')
    figure;
end;
mesh(xx,yy,zz);%каркасная поверхность функции f(x)
hold on;
extra = 0;
%траектория системы
plot3(x(:,1),x(:,2),z + extra,'b');

```

```

% траектория системы, спроектированная на поверхность f(x)
plot(x(:,1),x(:,2),'m');
%векторное поле, определяемое градиентом
quiver(-dx,-dy,'k');
%контурные линии, на которых f=const
contour(xx,yy,zz,'-c');
axis([xmin,xmax,ymin,ymax,zmin,zmax]);
title('Градиентные системы')
legend('каркасная поверхность функции f(x)',...
траектория системы','проекция на поверхность f(x)',...
'векторное поле, определяемое градиентом',...
'контурные линии, на которых f=const',...
'Location','SouthOutside');
grid on;
hold off;

```

Построение комплексных (алгебраических) фракталов

```

function varargout = complexfractal(varargin)
%
function btnPlot_Callback(hObject, eventdata, handles)
% Определение номера выбранной строки
Num = get(handles.pmType, 'Value'); if get(handles.cbInd,'Value'),
flag1=1;else flag1=0;end;
% Установка требуемого цвета линии
switch Num case 1
z0      = str2double(get(handles.EdtL,'String'));
z1      = str2double(get(handles.EdtR,'String'));
maxsteps= str2double(get(handles.EdtStep,'String'));
hres    = str2double(get(handles.EdtRes,'String'));
mandel(z0,z1,maxsteps,hres,flag1);
case 2
z0      = str2double(get(handles.EdtL,'String'));

```



```

z1      = str2double(get(handles.EdtR,'String'));
maxsteps= str2double(get(handles.EdtStep,'String'));
hres    = str2double(get(handles.EdtRes,'String'));
c       = str2double(get(handles.EdtC,'String'));
julia(c,z0,z1,maxsteps,hres,flag1);
case 3
    n= str2double(get(handles.EdtStep,'String'));
    newton(n);
end

```

Построение множества Жюлиа

```

function julia(c,z0,z1,maxsteps,hres,fla)
%c          - начальный параметр
%z0        - координата левой нижней точки
%z1        - координата правой верхней точки
%maxsteps  - число шагов для вычисления
%hres      - дискретность по оси x
x0 = real(z0);
y0 = imag(z0);
x1 = real(z1);
y1 = imag(z1);
    vres = ceil( abs( hres * (y1-y0)/(x1-x0) ) );
    dx = (x1-x0)/hres;
    dy = (y1-y0)/vres;
    [X,Y] = meshgrid(x0:dx:x1, y0:dy:y1);
    z = X + Y*i;
c = c*ones(size(z)); pict = 0*z; clear X Y for k=1:maxsteps
    bigs = find(abs(z)>=2);
    pict(bigs) = k*ones(size(bigs));
    z(bigs) = zeros(size(bigs));
    c(bigs) = zeros(size(bigs));
    z = z.^2 + c;

```

```

end; if (fla==1), figure;end;
    image(pict)
    axis('image')
    title('Множество Жюлиа')
    axis('xy')
    axis('off')

```

Построение множества Мандельброта

```

function mandel(z0,z1,maxsteps,hres,fla)
%z0      - координата левой нижней точки
%z1      - координата правой верхней точки
%maxsteps - число шагов для вычисления
%hres    - дискретность по оси x
    x0 = real(z0);
    y0 = imag(z0);
    x1 = real(z1);
    y1 = imag(z1);
    vres = ceil( abs( hres * (y1-y0)/(x1-x0) ) );
    dx = (x1-x0)/hres;
    dy = (y1-y0)/vres;
    [X,Y] = MESHGRID(x0:dx:x1, y0:dy:y1);
    c = X + Y*i;
    z = c;
    pict = 0*z;
    clear X Y
    for k=1:maxsteps
        bigs = find(abs(z)>=2);
        pict(bigs) = k*ones(size(bigs));
        z(bigs) = zeros(size(bigs));
        c(bigs) = zeros(size(bigs));
        z = z.^2 + c;
    end;

```

```

    if (fla==1), figure;end;
        image(pict)
        title('Множество Мандельброта')
    % colormap(hsv(maxsteps));
    axis('image')
    axis('xy')
    axis('off')

```

Построение множества Ньютона

```

function newton(n)
    n %количество шагов
    dx = 3/n; nits = 10; tol = 0.25; w1 = 1; w2 = exp(2*pi*i/3);
    w3=w2*w2;
    %Генерация матрицы...
    [X,Y] = meshgrid(-1.5:dx:1.5); m = X + i*Y;
    %Матрица размера int2str(prod(size(m))) создана
    %выполняются int2str(nits) шагов
    for k=1:nits
        m = m - (m.^3-1)./(3*m.^2);
    end;
    % mm записываем самый близкий корень
    mm = 1*(abs(m-w1)<tol) + 2*(abs(m-w2)<tol) + 3*(abs(m-w3)<tol);
    % the errs hold which entries are not converged yet
    errs = find(mm==0); nerr = prod(size(errs));
    xtra = 0;
    %теперь мы заканчиваем вычислять те вхождения, которые должны все же
    [jj,kk] = size(mm); for j = errs'
        x = m(j);
        xx = 1*(abs(x-w1)<tol)+2*(abs(x-w2)<tol)+3*(abs(x-w3)<tol);
        while xx==0
            x = x - (x*x*x-1) / (3*x*x);
            xtra = xtra+1;

```

```

        xx=1*(abs(x-w1)<tol)+2*(abs(x-w2)<tol)+3*(abs(x-w3)<tol);
    end; % end while
    m(j) = x;
    mm(j) = xx;
end; % end for loops and if mm==0
% выводим рисунок на экран
image(mm); axis('image'); axis('off');
colormap(eye(3)); % for color monitors

```

Построение геометрических фракталов с помощью систем итерируемых функций

```

function varargout = IFSfractal(varargin)
%
function btnPlot_Callback(hObject, eventdata, handles)
%матрица-система итерируемых функций
ifs = dlmread(get(handles.EdtF,'String')); npts=
str2double(get(handles.EdtB,'String')); [r,c] = size(ifs); if c<6
error('Matrix must have 6 columns'); end;
% найти точки фрактала ...
pt = [0;0]; A = [ ifs(1,1:2); ifs(1,3:4) ]; b = ifs(1,5:6)';
% ... повторяя первую функцию 50 раз
for k=1:50
pt = A*pt+b; end;
% затем мы вычисляем вес каждой функции
detlist = zeros(1,r); for k=1:r
    detlist(k) = abs(ifs(k,1)*ifs(k,4) - ifs(k,2)*ifs(k,3));
end;
% если все детерминанты - нуль, устанавливаем detlist в весь 1's
if all(detlist < 3*eps)
    detlist = ones(1,r);
end;
% найдем все нули в detlist...

```

```

zlist = find(detlist < 3*eps); least =
min(detlist(find(detlist>3*eps)));
% ...и устанавливаем минимальные значения
detlist(zlist) = (least/10)* ones(size(zlist));
% преобразуем detlist в вектор вероятности
detlist = detlist/sum(detlist);
% и вычислите выбранный вектор
% вычисляя совокупные суммы
selector = cumsum(detlist);
% мы предвычисляем наши выборы
rlist = rand(1,npts); choice = zeros(1,npts); for k=1:npts
    choice(k) = min(find(rlist(k) < selector));
end;
pts = zeros(2,npts); % для хранения точек
for k=1:npts
    p = choice(k);
    A = [ifs(p,1:2); ifs(p,3:4)];
    b = ifs(p,5:6)';
    pt = A*pt+b;
    pts(:,k) = pt;
end; if get(handles.cbInd,'Value'), figure; end;
% рисуем точки
plot(pts(1,:),pts(2,:),'.'); str=strcat('IFS-фрактал:
',get(handles.EdtF,'String')); title(str); axis('off');

```

Приложение В

Таблица преобразований Лапласа

$F(s)$	$f(t)$	$F(s)$	$f(t)$
$\frac{1}{s-a}$	e^{at}	$\frac{1}{s^2(s-a)}$	$\frac{1}{a^2}(e^{at} - 1 - at)$
$\frac{1}{1+\tau s}$	$\frac{1}{\tau}e^{-\frac{t}{\tau}}$	$\frac{1}{(s-a)(s-b)^2}$	$\frac{e^{at} - [1 + (a-b)t]e^{bt}}{(a-b)^2}$
$\frac{1}{s(s-a)}$	$\frac{1}{a}(e^{at} - 1)$	$\frac{1}{(s-a)(s-b)(s-c)}$	$\frac{e^{at}}{(b-a)(c-a)} + \frac{e^{bt}}{(a-b)(c-b)} + \frac{e^{ct}}{(a-c)(b-c)}$
$\frac{1}{(s-a)^2}$	te^{at}	$\frac{1}{(s-a)^3}$	$\frac{1}{2}t^2e^{at}$
$\frac{1}{(s-a)(s-b)}$	$\frac{e^{at} - e^{bt}}{a-b}$	$\frac{s}{(s-a)(s-b)^2}$	$\frac{ae^{at} - [a + b(a-b)t]e^{bt}}{(a-b)^2}$
$\frac{b+cs}{s(s-a)}$	$-\frac{b}{a} + (c + \frac{b}{a})e^{at}$	$\frac{s}{(s-a)(s-b)(s-c)}$	$\frac{ae^{at}}{(b-a)(c-a)} + \frac{be^{bt}}{(a-b)(c-b)} + \frac{ce^{ct}}{(a-c)(b-c)}$
$\frac{s}{(s-a)^2}$	$(1+at)e^{at}$	$\frac{s}{(s-a)^3}$	$(t + \frac{1}{2}at^2)e^{at}$
$\frac{s}{(s-a)(s-b)}$	$\frac{ae^{at} - be^{bt}}{a-b}$	$\frac{1}{s(s^2+a^2)}$	$\frac{1}{a^2}(1 - \cos at)$
$\frac{b+cs}{s^2+a^2}$	$c \cos at + \frac{b}{a} \sin at$	$\frac{1}{s(s^2-a^2)}$	$\frac{1}{a^2}(\operatorname{ch} at - 1)$

$F(s)$	$f(t)$	$F(s)$	$f(t)$
$\frac{1}{s^2 + as + b^2}$		$\frac{s}{s^2 + as + b^2}$	
$\Delta = b^2 - \frac{a^2}{4} > 0$	$\frac{1}{\sqrt{\Delta}} e^{-\frac{at}{2}} \sin \sqrt{\Delta} t$	$\Delta = b^2 - \frac{a^2}{4} > 0$	$e^{-\frac{at}{2}} \left(\cos \sqrt{\Delta} t - \frac{a}{2\sqrt{\Delta}} \sin \sqrt{\Delta} t \right)$
$\Delta = b^2 - \frac{a^2}{4} < 0$	$\frac{1}{\sqrt{-\Delta}} e^{-\frac{at}{2}} \operatorname{sh} \sqrt{-\Delta} t$	$\Delta = b^2 - \frac{a^2}{4} < 0$	$\frac{e^{-\frac{at}{2}}}{2\sqrt{-\Delta}} \left(\operatorname{sh} \sqrt{-\Delta} t - \frac{a}{2} \operatorname{ch} \sqrt{-\Delta} t \right)$
$\Delta = b^2 - \frac{a^2}{4} = 0$	$t e^{-\frac{at}{2}}$	$\Delta = b^2 - \frac{a^2}{4} = 0$	$e^{-\frac{at}{2}} \left(1 - \frac{at}{2} \right)$
$\frac{b + cs}{s^2 - a^2}$	$c \operatorname{ch} at + \frac{b}{a} \operatorname{sh} at$		

Таблица В.1: Оригиналы для дробно-рациональных изображений