

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Ярославский государственный университет им. П. Г. Демидова
Кафедра математического моделирования

Д. В. Глазков

Пакеты прикладных математических программ

*Методические указания
к проведению лабораторных работ*

*Рекомендовано
Научно-методическим советом университета
для студентов, обучающихся по специальности
Прикладная математика и информатика*

Ярославль 2009

УДК 519.2
ББК В185я 73
Г52

*Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2009 года*

Рецензент
кафедра математического моделирования
Ярославского государственного университета им. П. Г. Демидова

Глазков, Д. В. Пакеты прикладных математических программ:
Г52 метод. указания к проведению лабораторных работ / Д. В. Глазков;
Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль: ЯрГУ, 2009. — 40 с.

В методических указаниях приведено описание лабораторных работ.
Предназначены для студентов третьего курса, обучающихся по специальности 010501 Прикладная математика и информатика (дисциплина «Пакеты прикладных математических программ», блок ДС), очной формы обучения.

УДК 519.2
ББК В185я 73

©Ярославский государственный
университет им. П. Г. Демидова, 2009

Содержание

Введение	4
1. Лабораторная работа №1. Устойчивость состояний равновесия динамических систем	7
2. Лабораторная работа №2. Вычисление ляпуновских показателей	14
3. Лабораторная работа №3. Решение линейных интегральных уравнений	23
Заключение	32
Список литературы	34
Приложение	37

Введение

Современное математическое исследование редко обходится без использования вычислительной техники. В некоторых областях науки обойтись без численных алгоритмов уже невозможно. Многие рутинные процедуры давно стали частью пользовательских библиотек, которые входят в состав сред программирования. Однако стандартные средства разработки приложений зачастую не могут в полной мере удовлетворить потребности специалистов-математиков. Нельзя приветствовать ситуацию, когда ради приближенного вычисления значения обычного интеграла приходится писать собственные функции, а задача элементарного символьного дифференцирования и вовсе способна поставить в тупик начинающего программиста.

Необходимость решения такого рода задач привела к развитию одного из важных направлений приложения вычислительной техники — систем символьной математики. Их главной особенностью по сравнению с традиционным использованием ПК — действиями с числами — является возможность оперировать с выражениями по определенным правилам преобразований.

Основы для развития таких систем были заложены в 1960-х годах Джоном Маккарти — разработчиком языка списков Lisp. Главным объектом в Lisp является имя элемента, а главной операцией — подстановка. Это привело в дальнейшем к появлению систем, позволяющих работать с формулами, в частности, системы R-Lisp и языка Reduce.

Развитие принципов, заложенных в Lisp-е, послужило созданию более современных и функциональных систем символьной математики, таких как Derive, Maple и Mathematica. На основе ядра системы Maple возможности производить символьные преобразования были также заложены в популярные пакеты MATLAB и MathCad, которые широко известны не в последнюю очередь благодаря удобному пользовательскому интерфейсу. Кроме этого, в перечисленных системах реализованы все стандартные средства привычного «вычислительного» программирования, но уже на языках «сверхвысокого» уровня, дополненные большим числом встроенных библиотек. Все это позволяет рассматривать их как наиболее универсальный инструмент для математического исследования.

Отметим ключевое отличие между системами Mathematica и Maple, с одной стороны, и MATLAB — с другой. Оно заключается в том, что их создатели изначально преследовали различные цели. Если Mathematica

и Maple, как в свое время Lisp, предназначены для символьных преобразований, то MATLAB «заточен» на достижение максимальной эффективности программирования операций с векторами и матрицами. «Векторизация кода» не только облегчает труд программиста, ускоряя процесс разработки, но и позволяет сделать код обозримее по сравнению с обычными «скалярными» языками, что снижает вероятность «человеческих» ошибок при решении сложных задач. Тем не менее Symbolic Math Toolbox пакета MATLAB позволяет выполнять символьные преобразования, а в системах Mathematica и Maple реализованы мощные и удобные механизмы работы со списками. Поэтому перечисленные пакеты являются универсальными инструментами.

Универсальные математические системы [1–8] позволяют компактно и быстро реализовать специализированные математические преобразования, как числовые, так и символьные, входящие в состав сложных алгоритмов, из любой области математики. Такими системами являются пакеты MATLAB, Mathematica, Maple, в меньшей степени MathCad из-за ограниченности возможностей программирования. Отметим, что у MATLAB существуют свободно распространяемые аналоги, такие как Octave, SciLab. Их главное и несомненное преимущество может быть передано словами Free License Software. Обратная сторона медали — отсутствие удобной пользовательской оболочки. Весь интерфейс Octave ограничивается текстовым редактором SciTE, графической оболочкой gnuplot и командной строкой. Правда, Scintilla Text Editor, где первое слово можно перевести как «собранный по крупицам», понимает семантику многих языков программирования. Пакет Octave полностью совместим с MatLab и позволяет выполнять несложные программы, написанные для MatLab, практически без редактирования.

Перечисленные универсальные математические системы используют языки интерпретируемого типа, сравнимые с Basic или Perl в том смысле, что созданные в них программы не требуют специальной стадии компиляции в последовательность машинных команд, сохраняемую в отдельном файле. Ядро пакета Mathematica (или другого пакета) анализирует и выполняет команды в пределах одной входной ячейки (строки ввода) независимо от команд в других ячейках (строках). В MATLAB реализована возможность создания сценариев в m-файлах, сравнимых с CGI-сценариями в Perl.

Предлагаемые лабораторные работы имеют целью приобретение навыков программирования в двух-трех различных пакетах. Представ-

ленные задания не являются чрезмерно простыми — для их выполнения недостаточно знания встроенных функций обширных математических библиотек. Но, с другой стороны, поставленные задачи не оказываются слишком сложными, поскольку многие подзадачи не требуют дополнительных усилий и могут быть решены обращением к соответствующей стандартной функции. Это касается, например, вычисления матрицы Якоби для векторного поля, символьного интегрирования или задачи на собственные значения. Таким образом, пакеты прикладных математических программ позволяют сконцентрироваться на содержательной части работы, не отвлекаясь на решение стандартных вспомогательных подзадач. Это приобретает особенную ценность в самостоятельных серьезных исследованиях в форме курсовых и дипломных работ, а также в дальнейшей деятельности некоторых выпускников.

Для получения зачета по курсу «Пакеты прикладных математических программ» вместо одной или двух лабораторных работ допускается сдача программы, написанной в одном из пакетов для решения задачи, поставленной перед студентом в рамках курсовой работы.

1. Лабораторная работа №1. Устойчивость состояний равновесия динамических систем

Ставится задача нахождения состояний равновесия динамических систем, зависящих от параметров, и исследования их на устойчивость.

Рассматривается система обыкновенных дифференциальных уравнений, которая в простейшем случае в отсутствие параметров представима в виде

$$\dot{x} = F(x), \quad (1)$$

где $x = x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$ — вектор-функция, действующая из \mathbb{R} в \mathbb{R}^n , $F(x) = (F_1(x), F_2(x), \dots, F_n(x))^T$ — векторное поле из \mathbb{R}^n в \mathbb{R}^n .

Постоянная векторная величина $x^* \in \mathbb{R}^n$, которая обращает в нуль правую (а также левую как производную от константы) часть (1)

$$F(x^*) = 0, \quad (2)$$

называется состоянием равновесия системы (1).

Локальная динамика задачи (1) в малой окрестности состояния равновесия x^* определяется линеаризованной системой

$$\dot{u} = A u, \quad (3)$$

с постоянной матрицей

$$A = \left. \frac{DF}{Dx} \right|_{x=x^*} = \left(\begin{array}{cccc} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n} \end{array} \right) \bigg|_{x_k=x_k^*, \quad k=1,2,\dots,n}, \quad (4)$$

которая получается как матрица Якоби векторного поля F в точке x^* .

Устойчивость состояния равновесия системы (1) определяется из анализа расположения собственных значений λ_k матрицы A на комплексной плоскости. Именно, верна следующая

Теорема 1. Пусть $\forall k=1, \dots, n$ выполняется соотношение $\operatorname{Re} \lambda_k < 0$, тогда состояние равновесия x^* системы (1) устойчиво. Пусть $\exists k$ такое, что $\operatorname{Re} \lambda_k > 0$, тогда состояние равновесия x^* системы (1) неустойчиво.

В общем случае правая часть системы ОДУ зависит от параметров:

$$\dot{x} = F(x, \mu), \quad (5)$$

где x — по-прежнему вектор-функция из \mathbb{R} в \mathbb{R}^n , $\mu = (\mu_1, \mu_2, \dots, \mu_m)^T$ — вектор из \mathbb{R}^m , состоящий из параметров задачи. Система (5) имеет фазовое пространство (пространство переменных) размерности n и пространство параметров размерности m . При этом состояния равновесия системы (5), их число и устойчивость, вообще говоря, зависят от параметров задачи:

$$F(x^*, \mu) = 0, \quad x^* = x^*(\mu), \quad A = A(\mu), \quad \lambda_k = \lambda_k(\mu).$$

Зависимость собственных значений матрицы A от параметров может быть довольно сложной, поэтому зачастую оказывается удобнее оценить области устойчивости x^* в пространстве параметров, воспользовавшись критерием Рауса–Гурвица (см., например, [9, с. 133] или [10, с. 40]) или методом D -разбиений. В этом случае вместо функции вычисления собственных значений матрицы следует воспользоваться функцией, выдающей характеристический полином, коэффициенты которого, как правило, зависят от параметров задачи более простым образом.

Критерий Рауса–Гурвица (Raus, Hurwitz) формулируется так. Для оценки расположения корней характеристического уравнения

$$P(\lambda) = a_0\lambda^n + a_1\lambda^{n-1} + \dots + a_{n-1}\lambda + a_n = 0, \quad (6)$$

где, вообще говоря, коэффициенты $a_k = a_k(\mu)$ зависят от параметров системы (5), вводится в рассмотрение матрица Гурвица

$$H = \begin{pmatrix} a_1 & a_0 & 0 & 0 & \dots & 0 \\ a_3 & a_2 & a_1 & a_0 & \dots & 0 \\ a_5 & a_4 & a_3 & a_2 & \dots & 0 \\ a_7 & a_6 & a_5 & a_4 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_n \end{pmatrix}. \quad (7)$$

Диагональными элементами матрицы H являются величины a_1, \dots, a_n . По ее столбцам записаны коэффициенты либо только с нечетными, либо только с четными номерами. Если $k > n$ или $k < 0$, то $a_k = 0$. Главные миноры матрицы H имеют вид

$$\Delta_1 = a_1, \quad \Delta_2 = \begin{vmatrix} a_1 & a_0 \\ a_3 & a_2 \end{vmatrix}, \quad \Delta_3 = \begin{vmatrix} a_1 & a_0 & 0 \\ a_3 & a_2 & a_1 \\ a_5 & a_4 & a_3 \end{vmatrix}, \quad \dots, \quad \Delta_n = \det(H),$$

где как вертикальными чертами, так и буквами \det обозначен определитель соответствующей матрицы.

Справедлива следующая

Теорема 2. Пусть $a_0 > 0$. В этом случае все корни характеристического уравнения (6) с вещественными коэффициентами удовлетворяют условию $\operatorname{Re} \lambda_k < 0$ тогда и только тогда, когда выполнены неравенства

$$\Delta_1 > 0, \quad \Delta_2 > 0, \quad \dots, \quad \Delta_n > 0.$$

Отметим, что необходимым условием устойчивости состояния равновесия x^* системы (5) является положительность всех коэффициентов $a_k > 0$ уравнения (6).

Теорема 2 дает достаточно простой и удобный способ определения характера устойчивости состояния равновесия системы ОДУ по коэффициентам характеристического уравнения без явного вычисления его корней. Оценить преимущества такого подхода можно уже при $n=2$ и $n=3$.

Однако критерием Рауса–Гурвица не всегда удобно пользоваться. Если величина n велика, то конечные формулы становятся излишне громоздкими. Кроме того, критерий Рауса–Гурвица неприменим, если левая часть уравнения (6) не является полиномом — например, для систем с запаздыванием.

Метод D -разбиений лишен указанных недостатков. Идея метода состоит в разбиении пространства параметров на области D_k с различным числом $D(\mu)=k$ корней уравнения (6) в правой комплексной полуплоскости. Точка $\mu_* \in D_k$, если $D(\mu_*)=k$. Границы между указанными областями задаются из условия $\lambda=i\omega$, где ω есть вещественная величина, то есть из условия наличия корня на мнимой оси. Параметрическое уравнение границы между соседними областями D_{k_1} и D_{k_2} из \mathbb{R}^m задается соотношением $P(i\omega)=0$, где $-\infty < \omega < \infty$.

Задача определения области устойчивости состояния равновесия x^* в пространстве параметров сводится, таким образом, к нахождению D_0 . После того как все границы между областями построены, остается определить число корней k в правой комплексной полуплоскости для каждой из областей. Поскольку в отдельно взятой области оно не меняется, можно определить его в одной точке, при конкретном значении вектора μ . Примеры использования метода приведены в [9, 10].

Предлагается исследовать свойства системы (5) в соответствии со следующим алгоритмом.

1. При необходимости привести задачу к стандартному виду (5). Это нужно сделать, например, в случае присутствия в системе уравнения со второй производной.
2. Найти состояния равновесия $x^*(\mu)$. Выбрать одно из них для изучения устойчивости, если явно не указано какое. Приветствуется исследование на устойчивость всех найденных состояний равновесия.
3. С помощью одного из математических пакетов построить матрицу Якоби и матрицу A в соответствии с соотношением (4).
4. Получить характеристический полином и по его коэффициентам построить главные миноры из условий Рауса–Гурвица или, воспользовавшись методом D -разбиений, перейти к уравнению $P(i\omega)=0$.
5. На плоскости, определяемой двумя параметрами системы, построить границы областей устойчивости и указать эти области. Если параметров больше двух, то часть параметров положить равными некоторым константам. Возможно построение границ областей устойчивости в пространстве трех параметров, если такие графики окажутся визуально информативными.

В заключение проверить сделанные выводы об устойчивости состояния равновесия, численно решая систему (5) с начальными условиями, взятыми вблизи положения равновесия при конкретных значениях параметров.

Предлагаемые системы ОДУ с трехмерным фазовым пространством ($n=3$) являются математическими моделями различных реальных физических, химических и биологических процессов. Подробнее ознакомиться с происхождением моделей, смыслом переменных и параметров можно в цитируемой литературе.

1. Система уравнений Лоренца [11] (см. также [12–15]) моделирует тепловую конвекцию в нагреваемой жидкости и динамику многих типов лазеров

$$\begin{cases} \dot{x} = -\sigma(x-y), \\ \dot{y} = -y + (r-z)x, \\ \dot{z} = -bz + xy. \end{cases} \quad (8)$$

2. Замены переменных, параметров и времени [12, с. 366] позволяют преобразовать ее к виду

$$\begin{cases} \dot{x} = -\alpha x + y, \\ \dot{y} = -\beta y + xz, \\ \dot{z} = \gamma - z - xy, \end{cases} \quad (9)$$

3. а при условии $r > 1$ – к виду [13, с. 138]

$$\begin{cases} \ddot{x} + \delta \dot{x} + x^3 + (z-1)x = 0, \\ \dot{z} = -az - dx^2. \end{cases} \quad (10)$$

4. Циркуляционные движения неравномерно нагреваемой жидкости также могут быть адекватно описаны с помощью моделей Мура–Шпигеля [14, с. 79]

$$\begin{cases} \ddot{x} = z - (1-\delta)x, \\ \dot{z} = -\rho z + (1-\delta x^2)\dot{x}, \end{cases} \quad (11)$$

5. и Мариока–Шимицу [16]

$$\begin{cases} \dot{x} = y, \\ \dot{y} = x - ay - xz, \\ \dot{z} = -bz + x^2. \end{cases} \quad (12)$$

6. Химическая реакция Белоусова–Жаботинского описывается на языке дифференциальных уравнений [12, с. 319] следующим образом:

$$\begin{cases} \dot{x} = s(x + y - xy - qx^2), \\ \dot{y} = -y - xy - fz^2, \\ \dot{z} = w(x - z). \end{cases} \quad (13)$$

7. Модель Рёсслера химической реакции в ёмкости с перемешиванием [17], см. также [13, с. 116], имеет вид

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + \alpha y, \\ \dot{z} = b + (x - c)z. \end{cases} \quad (14)$$

8. Классический генератор с инерционной нелинейностью [13, с. 141], который в [15] назван генератором Анищенко–Астахова, моделируется системой

$$\begin{cases} \dot{x} = mx + y - xz, \\ \dot{y} = -x, \\ \dot{z} = -gz + gx^2, \end{cases} \quad (15)$$

9. а его модификация [13, с. 143] – уравнениями

$$\begin{cases} \dot{x} = mx + y - xz - dx^3, \\ \dot{y} = -x, \\ \dot{z} = -gz + g(e^x - 1). \end{cases} \quad (16)$$

10. Модель кольцевого генератора Дмитриева–Кислова [15, с. 89] имеет вид

$$\begin{cases} T\dot{x} = -x + Mze^{-x^2}, \\ \dot{y} = x - y, \\ \dot{z} = y - z/Q. \end{cases} \quad (17)$$

11. В [13, с. 138] приводится модель релаксационного генератора

$$\begin{cases} \ddot{x} - 2h\dot{x} + (1+\alpha)x + \alpha(z - z^3), \\ \dot{z} = \beta(x + z - z^3). \end{cases} \quad (18)$$

12. Динамика радиотехнического автогенератора хаотических колебаний Кияшко–Пиковского–Рабиновича с туннельным диодом [15, с. 86], называемого также в [9, с. 472] генератором шума, может быть описана системой ОДУ

$$\begin{cases} \dot{x} = 2hx + y - gz, \\ \dot{y} = -x, \\ \mu\dot{z} = x - f(z), \end{cases} \quad (19)$$

где функция $f(z) = 14z^3 - 22z^2 + 9z$.

13. В [9, с. 481] для упрощенного описания взаимодействия трех синхронизированных волн предлагается следующая система:

$$\begin{cases} \dot{x} = x + \delta y + z - 2y^2, \\ \dot{y} = -\delta x + y + 2xy, \\ \dot{z} = -2z(x+\nu). \end{cases} \quad (20)$$

14. Отметим близкие к (20) по духу модели колебательных систем, из которых одна предложена Рабиновичем и Фабрикантом и приведена в [13, с. 109]

$$\begin{cases} \dot{x} = (z-1+x^2)y + \gamma x, \\ \dot{y} = (3z+1-x^2)x + \gamma y, \\ \dot{z} = -2z(v+xy), \end{cases} \quad (21)$$

15. а другая – Хаяси и Каваками, см. [13, с. 111] (положить параметр $B=0$)

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -(x^2+3z^2)x/8 - k_1 y + B \cos t, \\ \dot{z} = -k_2(3x^2+z^2)z/8 + k_3. \end{cases} \quad (22)$$

16. Система Мэя–Леонарда [18] моделирует динамику численности трех взаимодействующих биологических популяций (исследовать на устойчивость состояние равновесия $x=y=z \neq 0$)

$$\begin{cases} \dot{x} = (1 - x - \alpha y - \beta z) x, \\ \dot{y} = (1 - \beta x - y - \alpha z) y, \\ \dot{z} = (1 - \alpha x - \beta y - z) z. \end{cases} \quad (23)$$

17. Динамика излучения одномодового лазера стоячей волны класса В с синусоидальной решеткой инверсии при слабом насыщении активной среды может быть описана системой трех дифференциальных уравнений [19, с. 98]

$$\begin{cases} \dot{x} = Gx(y + z - 1), \\ \dot{y} = A - y(1+x) - xz, \\ \dot{z} = -z(1+x) - xy/2. \end{cases} \quad (24)$$

18. Наиболее простая балансная двухуровневая модель лазера с нелинейным фильтром допускает запись в форме [19, с. 287]

$$\begin{cases} \dot{x} = Gx(y + z - 1), \\ \dot{y} = A - y(1+x), \\ \dot{z} = \delta B - z(\rho x + \delta). \end{cases} \quad (25)$$

Упражнение. Показать, что система (8) заменами переменных, параметров и времени может быть приведена к виду (9), а при $r > 1$ — к виду (10).

2. Лабораторная работа №2. Вычисление ляпуновских показателей

Рассмотрим вновь систему ОДУ (1), дополненную начальными условиями

$$\dot{x} = F(x), \quad x(0) = x_0.$$

Геометрическим образом решения $x_*(t)$ задачи Коши в фазовом пространстве является фазовая траектория, своя для каждого начального условия. Отдельные траектории могут со временем неограниченно приближаться к притягивающим инвариантным множествам, называемым аттракторами. Информация о такого рода множествах, как правило, и является результатом исследования динамических систем, которые выступают математическими моделями реальных физических процессов. В дальнейшем на основе этой информации оказывается возможно делать обоснованные выводы о механизмах, лежащих в основе тех или иных явлений.

Важной инвариантной характеристикой установившихся режимов динамических систем является спектр ляпуновских показателей. Эти вещественные величины обобщают характеристические значения состояний равновесия и мультипликаторы предельных циклов.

Показателем Ляпунова функции $\Phi(t)$ называется действительное число (см [13])

$$\Lambda[\Phi(t)] = \overline{\lim}_{t \rightarrow \infty} [t^{-1} \ln \|\Phi(t)\|]. \quad (26)$$

Определение (26) позволяет оценивать скорость возрастания (убывания) функции при больших t в сравнении с экспонентой.

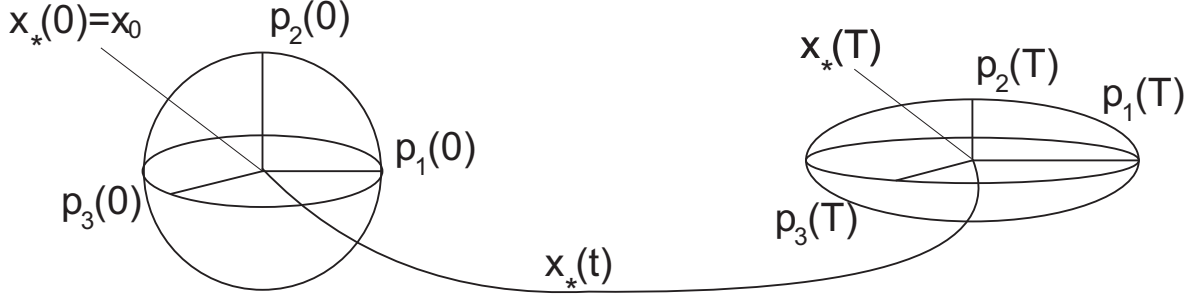


Рис. 1. Схематическая иллюстрация определения Каплана-Йорке и метода Бенеттина вычисления ляпуновских показателей

Классическое определение ляпуновских показателей нетривиальных решений систем ОДУ (а также дискретных отображений) связано с именами Каплана и Йорке [20]. Введем в рассмотрение шар с центром в точке x_0 сколь угодно малого радиуса в фазовом пространстве динамической системы, как показано на рисунке (1). Главные оси шара обозначим $p_j(0)$, $j=\overline{1, n}$, где n – размерность фазового пространства, а $V_n(0)$ – его объем. Взяв начальные условия в этом шаре, через время t под действием динамической системы мы попадем в некоторый эллипсоид с главными осями $p_j(t)$. Изменение величин главных осей с течением времени по аналогии с (26) и определяет ляпуновские показатели

$$\lambda_j = \overline{\lim}_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|p_j(t)\|}{\|p_j(0)\|}, \quad j=\overline{1, n}. \quad (27)$$

Будем считать, что $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, то есть эти величины упорядочены по убыванию. Объем первоначального n -мерного шара $V_n(0)$ изменяется со временем согласно закону

$$V_n(t) \simeq V_n(0) \exp[(\lambda_1 + \dots + \lambda_n)t].$$

Расстояние между двумя траекториями с близкими начальными условиями (из шара) изменяется в соответствии с соотношением

$$V_1(t) \simeq V_1(0) \exp[\lambda_1 t].$$

Это означает, что в случае $\lambda_1 > 0$ близкие траектории разбегаются, то есть имеет место локальная неустойчивость. Локальная, поскольку указанные законы действуют лишь в предположении малости величин $V_1(t)$, $V_1(0)$ или $V_n(t)$, $V_n(0)$.

Для диссипативных систем, к которым относится большинство реальных физических систем, характерным свойством является сжатие

фазового объема $V_n(t)$, которое выражается неравенством

$$\lambda_1 + \lambda_2 + \dots + \lambda_n < 0. \quad (28)$$

Соотношение (28) выражает также уменьшение полной энергии системы с течением времени, часть которой рассеивается или тратится на преодоление внешних сил (например, трения). Иными словами происходит диссипация (лат. *dissipatio* – рассеяние) или переход энергии упорядоченных процессов в энергию неупорядоченных. В отсутствие внешних сил, то есть при выполнении условия замкнутости, полная энергия системы сохраняется. С течением времени не изменяется также и фазовый объем, что означает отсутствие диссипации. Системы с такими свойствами называются консервативными (лат. *conservatio* – сохранение). Пренебрегая внешними силами, некоторые физические процессы, например, колебания маятника можно описать с помощью консервативных моделей. Однако даже малая диссипация с течением времени приводит к заметным качественным эффектам.

Укажем на возможность ситуации, когда наряду с неравенством (28) выполнено условие положительности старшего ляпуновского показателя $\lambda_1 > 0$. В этом случае говорят о сложном поведении, характерном для режимов так называемого динамического хаоса. Для их описания используют различные инвариантные характеристики, одной из которых является ляпуновская размерность. Вводится она следующим образом. В силу (28) и $\lambda_1 > 0$ существует такое число k , что

$$\lambda_1 + \lambda_2 + \dots + \lambda_k \geq 0 \quad \text{и} \quad \lambda_1 + \lambda_2 + \dots + \lambda_{k+1} < 0.$$

Очевидно, $\lambda_{k+1} < 0$. Тогда найдется такое число $0 \leq s < 1$, что

$$\lambda_1 + \lambda_2 + \dots + \lambda_k + s\lambda_{k+1} = 0.$$

Ляпуновской размерностью называют число

$$d_L = k + s = k + |\lambda_{k+1}|^{-1}(\lambda_1 + \lambda_2 + \dots + \lambda_k). \quad (29)$$

Для хаотических аттракторов величина d_L , вообще говоря, не является целой. Таким образом, фазовые траектории системы стремятся заполнить сложно устроенную область, размерность которой не удается охарактеризовать целым числом. Это, в частности, означает, что в фазовом пространстве остаются незаполненные участки значительного размера, называемые в некоторых источниках словом лакуны. Собственно, мерой «степени заполнения» «нелакунной» области траекториями и является величина d_L .

Соотношение (29) позволяет оценить ряд характеристик сложных режимов. Дело в том, что d_L дает оценку сверху для хаусдорфовой размерности аттрактора и энтропии динамической системы [21].

Идея вычисления ляпуновских показателей как классическим алгоритмом Бенеттина [22], так и его модификацией – методом динамической перенормировки [23] – состоит в следующем. Наряду с исходной задачей (1) с соответствующими начальными условиями рассмотрим так называемую систему в вариациях

$$\dot{u} = A(t) u, \quad \text{где} \quad A(t) = \left. \frac{DF}{Dx} \right|_{x=x_*(t)}. \quad (30)$$

Система (30) возникает как результат двух простых соображений. Во-первых, фазовая траектория со сколь угодно близкими к x_0 начальными условиями, лежащими на главных осях, может быть представлена в виде

$$x(t) = x_*(t) + p_j(t) = x_*(t) + h u_j(t), \quad (31)$$

где $p_j(t)$ есть вектор сколь угодно малой длины, представимый как произведение малой скалярной величины $h \ll 1$ и вектора $u_j(t)$ длины порядка единицы. Во-вторых, выполнив подстановку (31) в исходную систему (1) и собрав слагаемые одного порядка малости по h , получим, что отклонение решения $x(t)$ от $x_*(t)$ с точностью до слагаемых более высокого порядка малости по h будет определяться динамикой системы (30). Подчеркнем, что этот факт справедлив, пока величины $\|p_j(t)\|$ остаются достаточно малыми, то есть $\|u_j(t)\|$ не должны быть слишком велики.

Однако на пути расчета ляпуновских показателей существуют определенные препятствия. Например, при $\lambda_1 > 0$ величина $\|u_1(t)\|$ растет как $e^{\lambda_1 t}$, так что сделанные предположения вскоре теряют силу. Кроме того при вычислении на ПК значение $\|u_1(t)\|$ быстро выйдет за пределы машинной точности. Также следует помнить, что наибольшее влияние на $u_j(t)$ оказывает $e^{\lambda_1 t}$, то есть все u_j со временем поворачиваются в направлении наибольшего растяжения $u_1(t)$.

Во избежание подобных неприятностей поступают следующим образом. Через некоторое время T выполняется ортогонализация, в результате которой восстанавливается перпендикулярность главных осей. Однако полученный к моменту T эллипсоид, а значит, и его объем не меняется. Далее вычисляется отношение длин ортогональных друг другу главных осей к длинам соответствующих главных осей исходного шара (равным единице). Таким образом, получается приближенное

значение ляпуновских показателей на отрезке длины T . Затем производится перенормировка системы $u_j(t)$, в результате которой вновь имеем $\|u_j(T)\|=1$, то есть полученный к моменту T эллипсоид преобразуется вновь к шару, но центром уже в $x_*(T)$. Далее процедура повторяется. Время T выбирается эмпирически. Оно не должно быть ни слишком большим, иначе выйдем за границы машинной точности, ни слишком малым, иначе сильно упадет скорость расчетов. На практике можно следовать такому правилу: выполнить расчет для $T_1=1$, затем в зависимости от полученного $\lambda_1(T_1)$ взять новое значение T_2 ($T_2 > T_1$ для больших λ_1 , $T_2 < T_1$ для малых), снова произвести вычисления и сравнить ляпуновские показатели для T_1 и T_2 . Число m повторений основной процедуры выбирается так, чтобы расчет производился на достаточно большом интервале $t=mT$, величину которого на практике обычно берут от нескольких сотен до нескольких тысяч единиц.

Важной особенностью алгоритма является предположение, что начальные условия x_0 принадлежат аттрактору исходной системы (1). Это необходимо, поскольку измеряются характеристики аттрактора – инвариантного притягивающего множества в фазовом пространстве системы (1). Учесть указанную особенность можно, если рассчитать фазовую траекторию с произвольным начальным условием на достаточно большом промежутке времени. Конечная точка такого участка траектории уже должна располагаться близко к аттрактору так, что ее можно взять в качестве начального условия x_0 .

Обозначим $u_j^{(k)} = u_j(kT)$ и, пользуясь определением (27), преобразуем логарифмическую составляющую в момент времени $t=mT$

$$\begin{aligned} \ln \frac{\|p_j(t)\|}{\|p_j(0)\|} &= \ln \frac{\|u_j(t)\|}{\|u_j(0)\|} = \ln \left[\frac{\|u_j^{(1)}\|}{\|u_j^{(0)}\|} \cdot \frac{\|u_j^{(2)}\|}{\|u_j^{(1)}\|} \cdot \dots \cdot \frac{\|u_j^{(m)}\|}{\|u_j^{(m-1)}\|} \right] = \\ &= \ln \frac{\|u_j^{(1)}\|}{\|u_j^{(0)}\|} + \ln \frac{\|u_j^{(2)}\|}{\|u_j^{(1)}\|} + \dots + \ln \frac{\|u_j^{(m)}\|}{\|u_j^{(m-1)}\|}, \end{aligned}$$

так что в результате формула (27) примет вид

$$\lambda_j = \overline{\lim}_{m \rightarrow \infty} \frac{1}{mT} \sum_{k=0}^{m-1} \ln \frac{\|u_j^{(k+1)}\|}{\|u_j^{(k)}\|}, \quad j = \overline{1, n}. \quad (32)$$

В качестве главных осей $p_j(0)$ берутся единичные орты e_j , умноженные на величину h . Векторы e_j являются начальными условиями системы (30), так что $u_j^{(0)} = e_j$.

Для численного расчета спектра ляпуновских показателей нужно решать систему из $n+n^2$ уравнений. Исходная система состоит из n уравнений, и система в вариациях с n различными начальными условиями даст еще n^2 уравнений.

Опишем алгоритм расчета спектра ляпуновских показателей.

1. Задать ортонормированный базис (ОНБ) из n векторов $v_j^{(0)}$ (обычно $v_j^{(0)} = e_j$), начальное значение суммы $\sigma_j := \sigma_j(0) = 0$, шаг перенормировки T . Положить текущее число перенормировок $k=0$.
2. Решить систему из $n+n^2$ ОДУ на отрезке $[kT, (k+1)T]$. Система включает n уравнений (1) с начальными условиями $x(kT) = x_*(kT)$, получаемыми на предыдущем шаге, а также n уравнений (30), взятых n раз с различными начальными условиями $u_j^{(k)} = v_j^{(k)}$.

3. Перейти от старой системы векторов $u_j^{(k+1)}$ к новой $w_j^{(k+1)}$, выполнив ортогонализацию Грама–Шмидта (без нормировки)

$$\begin{aligned} w_1^{(k+1)} &= u_1^{(k+1)}, \\ w_2^{(k+1)} &= u_2^{(k+1)} - (u_2^{(k+1)}, w_1^{(k+1)}) w_1^{(k+1)}, \\ &\dots \\ w_n^{(k+1)} &= u_n^{(k+1)} - (u_n^{(k+1)}, w_{n-1}^{(k+1)}) w_{n-1}^{(k+1)} - \dots - (u_n^{(k+1)}, w_1^{(k+1)}) w_1^{(k+1)}. \end{aligned}$$

4. Увеличить текущее значение суммы σ_j на величину $\ln \|w_j^{(k+1)}\|$. Получим $\sigma_j := \sigma_j(k+1) = \sigma_j(k) + \ln \|w_j^{(k+1)}\|$. В качестве $(k+1)$ -го приближения j -го ляпуновского показателя выступает величина

$$\lambda_j \approx \sigma_j(k+1) / [(k+1)T].$$

5. Нормировать систему векторов $w_j^{(k+1)}$ и получить ОНБ на следующем шаге

$$v_j^{(k+1)} = w_j^{(k+1)} / \|w_j^{(k+1)}\|.$$

Увеличить значение k на единицу.

6. Повторить m раз пункты 2 – 5.
7. В качестве m -го приближения j -го показателя выступает

$$\lambda_j \approx \frac{\sigma_j(m)}{mT} = \frac{\sigma_j}{mT}.$$

В том случае, если нужен только старший ляпуновский показатель, а знание всего спектра необязательно, представленный алгоритм упрощается. Во-первых, достаточно решать систему из $2n$ уравнений – n уравнений исходной задачи Коши и n уравнений системы в вариациях с каким-то начальным условием $v_1^{(0)}$ единичной нормы. Соответственно изменятся пункты 1 – 2. Во-вторых, пункт 3 опускается, величины $u_1^{(k+1)}$ и $w_1^{(k+1)}$ совпадают и в конечном счете нас интересует только значение $\sigma_1(m)$.

При вычислении ляпуновских показателей динамических систем с непрерывным временем оказывается полезным знание следующих их свойств.

Во-первых, если система (1) диссипативна и ее аттрактор не является состоянием равновесия, то $\exists j : \lambda_j = 0$. В случае устойчивого предельного цикла $\lambda_1 = 0$, что было впервые доказано А. А. Андроновым [13, 24]. В общем случае для произвольного ограниченного решения системы (1), не стремящегося к особой точке, найдется, по крайней мере, один нулевой показатель [13]. В самом деле, рассмотрим наряду с ограниченным решением $x_*(t)$ системы (1), принадлежащим аттрактору, функцию $x_*(t+\tau)$, которая также является решением автономной системы (1). В силу гладкости правой части системы (1) и ограниченности $x_*(t)$ имеем

$$\lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|x_*(t+\tau) - x_*(t)\|}{\|x_*(\tau) - x_*(0)\|} = 0. \quad (33)$$

Таким образом, в фазовом пространстве системы (1) существует направление, вдоль которого расстояние между траекториями остается в среднем неизменным.

Во-вторых, сумма ляпуновских показателей, как было отмечено выше, определяет скорость изменения фазового объема $V_n(t)$. С другой стороны, по формуле Остроградского–Лиувилля

$$V_n(t) \simeq V_n(0) \exp \left[\int_0^t \text{tr} A(\tau) d\tau \right] = V_n(0) \exp \left[\int_0^t \nabla F(x_*(t)) d\tau \right],$$

то есть величина $V_n(t)$ определяется дивергенцией векторного поля F вдоль траектории $x_*(t)$:

$$D(t) = \text{div} F(x_*(t)) = \nabla F(x_*(t)) = \sum_{k=1}^n \frac{\partial F_j(x)}{\partial x_j} \Big|_{x=x_*(t)} = \text{tr} A(t).$$

Если дивергенция постоянна $D(t) \equiv D$, то $\lambda_1 + \dots + \lambda_n = D$.

Для хаотических аттракторов в случае $n=3$ это означает, что выполняются соотношения $\lambda_2=0$ и $\lambda_1+\lambda_3=D$, что позволяет ограничиться только вычислением дивергенции и старшего ляпуновского показателя. Для систем более высокой размерности $n>3$ указанные свойства дают возможность проверять корректность результатов вычислений.

Для автономных динамических систем вида (1) имеет место оценка

$$\lambda_1 + \dots + \lambda_n \geq \overline{\lim}_{t \rightarrow \infty} \frac{1}{t} \int_0^t \text{tr} A(\tau) d\tau. \quad (34)$$

Точное равенство имеет место для правильных по Ляпунову систем.

Вычисление ляпуновских показателей возможно и для систем с запаздыванием [25, 26]

$$\dot{x} = F(x, x(t-h)), \quad (35)$$

у которых правая часть зависит не только от значения неизвестной функции $x=x(t)$ в текущий момент времени t , но и в некоторый предшествующий момент $t-h$. Для корректной постановки задачи системе вида (35) следует дополнить начальными условиями. Отметим, что значения $x(0)$ в начальной точке недостаточно. Задачу Коши применительно уравнениям типа (35) впервые ввел в рассмотрение Н. Н. Красовский [27]. Начальные условия здесь выбираются из некоторого класса функций, заданных на отрезке длины запаздывания, например, из класса непрерывных функций $C[-h, 0]$. Поскольку размерность фазового пространства динамической системы полагают равной размерности пространства начальных условий, то системы с запаздыванием оказываются бесконечномерными.

В этом случае система в вариациях имеет вид

$$\dot{u} = A(t) u + B(t) u(t-h), \quad (36)$$

где

$$A(t) = \left. \frac{DF}{Dx} \right|_{x=x_*(t)}, \quad B(t) = \left. \frac{DF}{Dx(t-h)} \right|_{x=x_*(t)}.$$

Вычисление всего спектра ляпуновских показателей для систем с запаздыванием оказывается невозможным в силу их бесконечного числа. Однако получение приближенного значения старшего из показателей с помощью описанного алгоритма возможно, и это широко используется на практике.

Предложенная последовательность действий для расчета ляпуновских показателей может быть использована и для определения свойств динамических систем с дискретным временем. Такие системы допускают преобразование к виду

$$x^{(k+1)} = f(x^{(k)}), \quad x^{(0)} = \xi, \quad (37)$$

где f — отображение, действующее из \mathbb{R}^n в \mathbb{R}^n , ξ — вектор с начальными условиями. Линейная система в вариациях вводится аналогично рассмотренным ситуациям:

$$u^{(k+1)} = A^{(k)} u^{(k)}, \quad \text{где} \quad A^{(k)} = \left. \frac{Df}{Dx} \right|_{x=x^{(k)}}. \quad (38)$$

Основное отличие от случая динамических систем с непрерывным временем заключается в следующем. В пункте 1 предложенного алгоритма естественным образом вводится шаг перенормировки, равный целому числу итераций, обычно $T=1$. В пункте 2 вычисляется следующая итерация отображения f в соответствии с (37) и n отображений Df/Dx с n различными начальными условиями в соответствии с (38). Все остальные этапы описанного алгоритма не претерпевают изменений.

Выполнять расчет ляпуновских показателей предлагается для систем ОДУ (8) – (25) из лабораторной работы №1. Большая часть из них имеет режимы, для которых $\lambda_1 > 0$. Предлагается самостоятельно найти значения параметров, при которых наблюдаются сложные аттракторы с $\lambda_1 > 0$. Сведения о них можно узнать из цитируемой литературы. Для тестирования программ рекомендуется воспользоваться модельными системами Спротта, представленными в [15, с. 78] и [28, с. 60].

Расчет ляпуновских показателей представляет интерес при определении меры предсказуемости поведения той или иной динамической системы. Величину $\tau_1 \sim 1/\lambda_1$, обратно пропорциональную старшему ляпуновскому показателю, принимают за горизонт предсказуемости данной системы. Оценка τ_1 позволяет определить границы достоверного прогноза поведения изучаемого объекта. Такого рода знания востребованы в работе гидрометеослужб, биржевых аналитиков, экономистов, социологов, гео- и астрофизиков.

3. Лабораторная работа №3. Решение линейных интегральных уравнений

Наряду с задачами, которые могут быть сформулированы на языке дифференциальных уравнений, существует широкий класс задач, описываемый в терминах интегральных уравнений. Их основная особенность состоит в том, что неизвестная функция находится под знаком интеграла. Такого рода задачи встречаются в различных физических и технических приложениях и представляют определенный интерес для математиков [29–34].

К интегральным уравнениям сводится широкий круг задач оптимизации (задача о таутохроне, изопериметрическая задача и другие вопросы вариационного исчисления), математической физики (описание свободных и вынужденных колебаний струны, колебаний стержня, мембраны, равновесие упругой нити с подвешенным к ней грузом, краевая задача Штурма–Лиувилля и так далее), а также вопросов, связанных с интегральными преобразованиями (Фурье, Лапласа и других).

Но, пожалуй, наиболее очевиден переход к интегральным уравнениям от обыкновенных дифференциальных уравнений. В основе всех численных методов решения систем ОДУ лежит операция интегрирования. Заметим, однако, что далеко не всякое интегральное уравнение может быть представлено как дифференциальное.

Рассмотрим следующее линейное дифференциальное уравнение второго порядка:

$$\ddot{x} + a_1(t)\dot{x} + a_2(t)x(t) = F(t), \quad (39)$$

где точкой обозначается операция дифференцирования по времени t , с начальными условиями

$$x(0) = C_0, \quad \dot{x}(0) = C_1. \quad (40)$$

Обозначим

$$\ddot{x}(t) = \varphi(t), \quad \text{тогда} \quad \dot{x}(t) = \int_0^t \varphi(s) ds + C_1. \quad (41)$$

Наконец, $x(t)$ определяется по формуле

$$x(t) = \int_0^t (t-s) \varphi(s) ds + C_1 t + C_0. \quad (42)$$

В соотношении (42) две возникающие операции интегрирования заменены одной по формуле

$$\int_0^t dt_n \int_0^{t_n} dt_{n-1} \dots \int_0^{t_2} f(t_1) dt_1 = \frac{1}{(n-1)!} \int_0^t (t-s)^{n-1} f(s) ds. \quad (43)$$

Упражнение. Доказать формулу (43) при $n=2, n=3$.

Таким образом, исходная задача Коши (39), (40) при учете соотношений (41), (42) сводится к интегральному уравнению Вольтерра второго рода

$$\varphi(t) = \int_0^t K(t, s) \varphi(s) ds + f(t), \quad (44)$$

где ядро

$$K(t, s) = -[a_1(t) + a_2(t)(t-s)],$$

а неоднородность

$$f(t) = F(t) - C_1 a_1(t) - C_1 t a_2(t) - C_0 a_2(t).$$

Аналогичным образом к интегральным уравнениям вида (44) легко свести дифференциальные уравнения более высокого порядка, чем (39).

Кроме того, формула (43) позволяет ввести в рассмотрение операцию интегрирования дробного порядка [33, 34]. В самом деле, в правой части (43) требование $n \in \mathbb{N}$ не является необходимым. Если заменить факториал на гамма-функцию, можно брать и нецелые значения n . В этом случае интеграл от функции $f(t)$ произвольного порядка $\alpha > 0$ в предположении $t > 0$ по определению равен

$$I_{0+}^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s) ds. \quad (45)$$

Эта величина носит наименование левостороннего интеграла Римана–Лиувилля [33] дробного порядка. Правосторонний интеграл Римана–Лиувилля в предположении $t < 0$ определяется аналогично

$$I_{0-}^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_t^0 (s-t)^{\alpha-1} f(s) ds. \quad (46)$$

Несколько сложнее оказывается ввести обратную операцию — операцию дифференцирования дробного порядка. Дело в том, что формулы (45), (46) корректны не при всех $\alpha < 0$ (например, при целых отрицательных α гамма-функция $\Gamma(\alpha)$ не определена). Однако при $\alpha \in (-1, 0)$ величины $I_{0+}^\alpha f(t)$ и $I_{0-}^\alpha f(t)$, вообще говоря, могут быть подсчитаны, поэтому левосторонняя и правосторонняя производная Римана–Лиувилля дробного порядка меньше единицы вводятся без труда. В общем случае при обозначении

$$\alpha = (m - 1) + \beta, \quad \text{где } m = [\alpha] + 1, \quad \beta = \{\alpha\}, \quad (47)$$

используются формулы

$$\begin{aligned} D_{0+}^\alpha f(t) &= \left(\frac{d}{dt}\right)^{m-1} D_{0+}^\beta f(t) = \left(\frac{d}{dt}\right)^m I_{0+}^{1-\beta} f(t) = \\ &= \frac{1}{\Gamma(1-\beta)} \left(\frac{d}{dt}\right)^m \int_0^t \frac{f(s)}{(t-s)^\beta} ds, \end{aligned} \quad (48)$$

для $t > 0$ и

$$\begin{aligned} D_{0-}^\alpha f(t) &= \left(-\frac{d}{dt}\right)^{m-1} D_{0-}^\beta f(t) = \left(-\frac{d}{dt}\right)^m I_{0-}^{1-\beta} f(t) = \\ &= \frac{(-1)^m}{\Gamma(1-\beta)} \left(\frac{d}{dt}\right)^m \int_t^0 \frac{f(s)}{(s-t)^\beta} ds, \end{aligned} \quad (49)$$

для $t < 0$. Таким образом, заключаем, что операция дифференцирования дробного порядка сводится к суперпозиции операций обычного целого дифференцирования и дробного интегрирования. Например, при отыскании величин $D_{0+}^{3.7} f(t)$ и $D_{0-}^{3.7} f(t)$, то есть производных порядка $\alpha = 3.7$, в формулах (48), (49) следует взять $m = 4$ и $\beta = 0.7$.

Отметим, что задача исследования дифференциальных уравнений дробного порядка, рассматриваемая в [33], также может быть сведена к изучению свойств интегральных уравнений вида (44).

Кинетическое описание фрактальных свойств движения базируется на дробном обобщении уравнения Фокера–Планка–Колмогорова [34]. Фрактальные свойства пространства и времени, обусловленные спецификой структуры резонансов, в этом случае оказывается возможно описать на языке дробного исчисления.

Интегральным уравнением Фредгольма второго рода называется уравнение вида

$$\varphi(t) = \int_a^b K(t, s) \varphi(s) ds + f(t), \quad (50)$$

где K и f — известные функции, а неизвестную функцию φ требуется найти. Переменные s и t принимают значения из отрезка $[a, b]$. Таким образом, функции φ и f определены на отрезке $[a, b]$, а K — на квадрате $[a, b] \times [a, b]$, показанном на рис. (2).

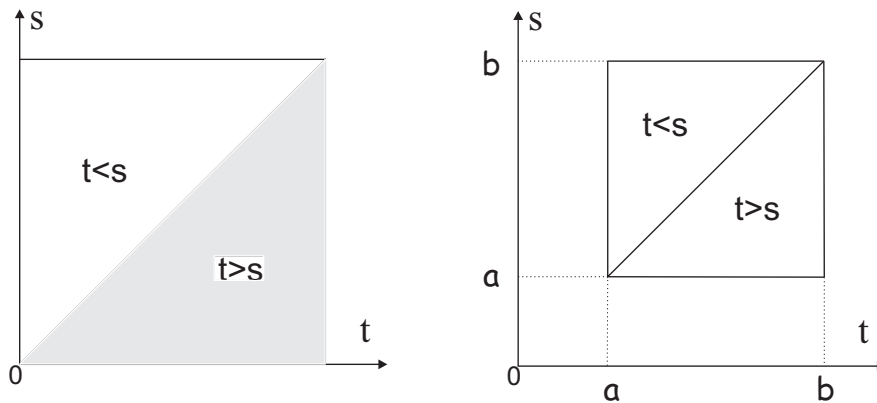


Рис. 2. Области определения ядер $K(t, s)$ линейных интегральных уравнений (44) и (50) слева и справа соответственно

Интегральные уравнения первого рода не содержат неизвестную функцию вне интеграла. Указанные типы уравнений представлены в таблице.

уравнение	Вольтерра	Фредгольма
1 рода	$\int_a^t K(t, s) \varphi(s) ds = f(t)$	$\int_a^b K(t, s) \varphi(s) ds = f(t)$
2 рода	$\varphi(t) = \int_a^t K(t, s) \varphi(s) ds + f(t)$	$\varphi(t) = \int_a^b K(t, s) \varphi(s) ds + f(t)$

Заметим, что класс интегральных уравнений Фредгольма включает в себя соответствующий класс уравнений Вольтерра. В самом деле, определив ядро уравнения Вольтерра $K(t, s) \equiv 0$ при $s > t$ (см. рис. (2)) и заменив верхний предел интегрирования t на b , можем перейти к уравнению Фредгольма. Обратный переход, вообще говоря, не всегда возможен.

Среди известных способов отыскания решения линейных интегральных уравнений отметим два.

Метод последовательных приближений. Суть этого метода заключается в построении последовательности функций, сходящейся к искомому решению $\varphi_n(t) \rightarrow \varphi(t)$ при $n \rightarrow \infty$. Такая последовательность, вообще говоря, не всегда будет сходящейся. Для этого должны быть выполнены определенные условия.

Рассмотрим семейство уравнений (50), зависящих от параметра λ :

$$\varphi(t) = \lambda \int_a^b K(t, s) \varphi(s) ds + f(t). \quad (51)$$

При $\lambda=0$ эта задача, очевидно, имеет единственное решение $\varphi(t)=f(t)$.

Введем в рассмотрение оператор A , действующий из пространства непрерывных на отрезке $[a, b]$ функций $C[a, b]$ в $C[a, b]$:

$$A\varphi = \lambda \int_a^b K(t, s) \varphi(s) ds + f(t). \quad (52)$$

Оператор A называют интегральным оператором Фредгольма с (непрерывным) ядром $K(t, s)$. При достаточно малых λ он является сжимающим в заданном функциональном пространстве, то есть существует такое $0 < \alpha < 1$, что в метрике этого пространства для любых элементов x, y справедливо соотношение

$$\rho(Ax, Ay) \leq \alpha \rho(x, y). \quad (53)$$

Например, в пространстве непрерывных на отрезке функций $C[a, b]$ расстояние между двумя элементами x, y вводится как

$$\rho(x, y) = \max_{a \leq t \leq b} |x(t) - y(t)|,$$

а в пространстве интегрируемых с квадратом функций $L_2[a, b]$ — так:

$$\rho(x, y) = \left(\int_a^b (x(t) - y(t))^2 dt \right)^{1/2}.$$

Вводя обозначения

$$M = \max_{a \leq t, s \leq b} |K(t, s)|, \quad B = \left(\int_a^b \int_a^b K^2(t, s) dt ds \right)^{1/2},$$

условие сжимаемости оператора A в $C[a, b]$ можем записать в виде

$$|\lambda| < 1/(M(b-a)),$$

а в $L_2[a, b]$ – в виде

$$|\lambda| < 1/B.$$

Упражнение. Обосновать приведенные оценки для λ .

В этом случае в силу известных теорем о сжимающих отображениях существует единственный элемент x_* такой, что

$$Ax_* = x_*.$$

Элемент x_* , который называют неподвижной точкой оператора A , или в нашем случае функция $\varphi_*(t)$, обладающая указанным свойством, очевидно, будет искомым решением интегрального уравнения (51).

Таким образом, когда выполняется условие (53), задача решения уравнения (51) сводится к построению последовательных приближений

$$\varphi_{n+1}(t) = A\varphi_n(s),$$

начиная с некоторой произвольной функции $\varphi_0(t)$, входящей в область определения оператора A . Отметим, что в случае уравнения Вольтерра второго рода соответствующий оператор всегда будет сжимающим.

Функцию $\varphi_{n+1}(t)$ будем называть приближенным решением с наперед заданной точностью ε , если выполнено соотношение

$$\rho(\varphi_{n+1}, \varphi_*) < \varepsilon.$$

Однако, поскольку точное решение заранее неизвестно, как правило, ограничиваются оценкой

$$\rho(\varphi_{n+1}, \varphi_n) < \varepsilon.$$

Выполнение этого неравенства является условием выхода из цикла, в котором реализуется итерационный процесс построения $\{\varphi_n\}$.

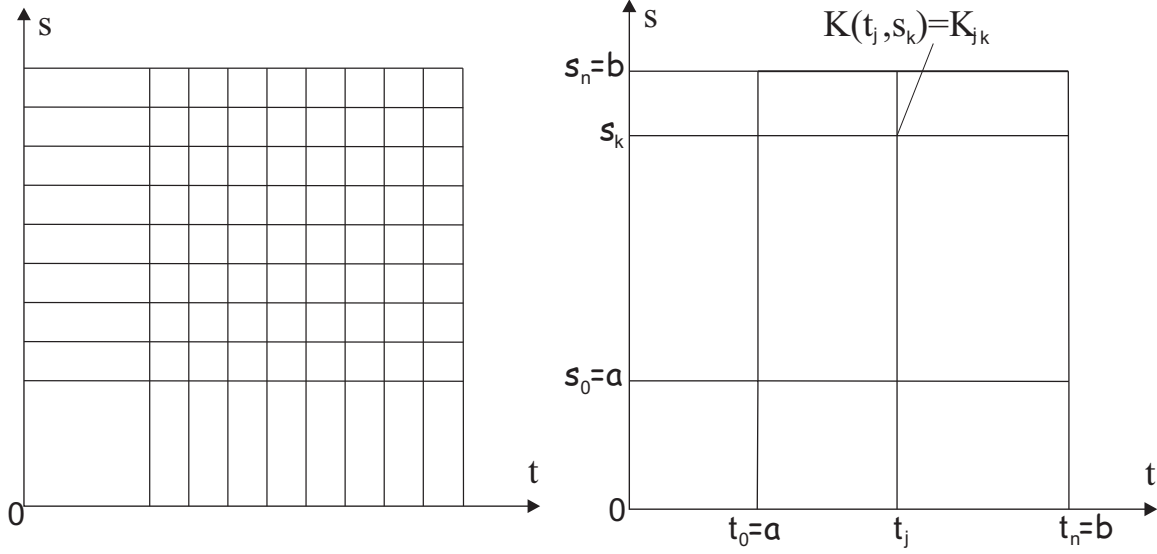


Рис. 3. Покрывтие квадрата сеткой и ее (j, k) -ый узел

Сведение к системе линейных алгебраических уравнений. В основе альтернативного метода приближенного решения задачи (51) лежит идея замены непрерывных функций K, φ, f их сеточными аппроксимациями.

Выбрав некоторое не слишком малое натуральное n , разбиваем отрезок $[a, b]$ на n равных частей, как показано на рисунке (3) так, что

$$t_j = s_j = a + j \frac{b-a}{n}, \quad \text{в частности,} \quad t_0 = s_0 = a, \quad t_n = s_n = b.$$

Далее приближенно (по формулам прямоугольника, трапеций, Симпсона и т.д.) вычисляем интеграл. Формула трапеций даст следующее уравнение:

$$\varphi(t) = \lambda \frac{b-a}{n} \sum_{k=1}^n \frac{1}{2} \left[K(t, s_{k-1}) \varphi(s_{k-1}) + K(t, s_k) \varphi(s_k) \right] + f(t). \quad (54)$$

Подставим сюда вместо t значения $t_j = s_j$, где $j=0, 1, \dots, n$. Обозначим

$$K_{jk} = K(s_j, s_k), \quad \varphi_j = \varphi(s_j), \quad f_j = f(s_j).$$

В результате получим систему из $1+n$ линейных алгебраических уравнений относительно $1+n$ неизвестных величин φ_j :

$$\varphi_j = \lambda \frac{b-a}{n} \left[\frac{1}{2} \left[K_{j0} \varphi_0 + K_{jn} \varphi_n \right] + \sum_{k=1}^{n-1} K_{jk} \varphi_k \right] + f_j, \quad j = 0, 1, \dots, n. \quad (55)$$

В векторной форме (jk -ый элемент матрицы Q зависит от K_{jk}) ее можно записать как

$$(I - Q)\varphi = f. \quad (56)$$

Решив эту систему, мы тем самым находим приближенные значения искомой функции в указанных узлах. Как правило, уже величины $n=100$ достаточно для построения хорошего графика.

Для своего варианта задания использовать два метода определения приближенного решения интегрального уравнения. Предварительно проверить условия сжимаемости оператора A при $\lambda=1$. Результаты представить в виде двух графиков и сравнить между собой.

Решить уравнение Фредгольма (задачи 1 – 9) или Вольтерра (задачи 10 – 18) второго рода:

1. $a = 0, \quad b = \pi, \quad K(t, s) = \frac{1}{\pi} \cos(t+s), \quad f(t) = 3 \sin(t)$
2. $a = 0, \quad b = 1, \quad K(t, s) = \frac{(t+s)^3}{1+s+s^2+s^3}, \quad f(t) = \frac{3}{4} - \frac{t^2}{2}$
3. $a = 0, \quad b = \frac{\pi}{5}, \quad K(t, s) = \frac{5}{\pi} \sin(5t+5s), \quad f(t) = \frac{3}{2} \sin(5t)$
4. $a = 0, \quad b = 1, \quad K(t, s) = \frac{\text{sh}(t+s)}{\text{ch}(1+s)}, \quad f(t) = \text{ch}(t)$
5. $a = 0, \quad b = 1, \quad K(t, s) = \frac{(t+s)^2}{1+s+s^2}, \quad f(t) = \frac{2}{3}$
6. $a = 0, \quad b = \pi, \quad K(t, s) = \frac{1}{\pi} \cos(3t+3s), \quad f(t) = 2 \cos(3t)$
7. $a = 0, \quad b = 1, \quad K(t, s) = s^4 - t^2 s, \quad f(t) = 5t^2 - 4/7$
8. $a = 0, \quad b = \pi, \quad K(t, s) = \frac{1}{\pi} \cos(t+s), \quad f(t) = \frac{1}{2} \cos(t)$
9. $a = 1, \quad b = 2, \quad K(t, s) = \frac{t^2}{s} - \frac{t}{s^2}, \quad f(t) = \frac{t}{2} - \frac{t^2}{4}$
10. $a = 0, \quad b = 1, \quad K(t, s) = -\text{sh}(t-s), \quad f(t) = t$
11. $a = 0, \quad b = \pi, \quad K(t, s) = \sin(t-s), \quad f(t) = t$

12. $a = 0, \quad b = \pi, \quad K(t, s) = 2 \cos(t-s), \quad f(t) = t \cos(t)$
13. $a = 0, \quad b = 1, \quad K(t, s) = s - t, \quad f(t) = t$
14. $a = 0, \quad b = 1, \quad K(t, s) = 1, \quad f(t) = 2 \cos(t)$
15. $a = 0, \quad b = \pi, \quad K(t, s) = e^{s-t} \sin(t-s), \quad f(t) = e^{-t}$
16. $a = 0, \quad b = 1, \quad K(t, s) = -e^{t-s}, \quad f(t) = t$
17. $a = 0, \quad b = \pi, \quad K(t, s) = e^{2(s-t)}, \quad f(t) = t(1+e^{-2t})$
18. $a = 0, \quad b = 1, \quad K(t, s) = e^{t-s}, \quad f(t) = 2.$

Заключение

При работе с математическими системами важно помнить, что при всем богатстве заложенных в них возможностей они далеко не безграничны, и всегда найдутся задачи, решение которых может потребовать отказа от стандартных средств в пользу разработки своих инструментов. Поэтому стоит быть готовым к написанию собственных функций в соответствии со всеми фундаментальными принципами методов вычислений. Вместе с тем важно иметь в виду, что можно проводить весьма серьезные исследования, пользуясь исключительно встроенными библиотеками прикладных математических программ. Образно говоря, если сравнивать «обычные» оболочки программирования с лопатой в сфере «численно-аналитического копания», то математические системы можно уподобить экскаватору, который в определенных ситуациях использовать предпочтительнее.

В целом книг [1–7] и информации, доступной в Интернете, достаточно для овладения основными приемами работы в перечисленных пакетах. Тем не менее весьма важным и полезным навыком, отличающим квалифицированного пользователя, является способность эффективно работать со встроенной справочной системой, которая содержит подробное описание стандартных функций и примеров их непосредственного практического использования. Грамотная работа со справочным материалом может принести больше пользы, чем чтение десятка книг. Например, обращение к описанию функции `NDSolve` численного решения систем дифференциальных уравнений в пакете `Mathematica 7` дает гораздо больше информации, чем обращение к «традиционным» источникам.

Одной из особенностей универсальных математических систем является свойство полиморфизма, реализованное через перегруженные функции (*overloaded function*). Полиморфная функция объединяет в себе несколько различных функций с общим заголовком, но разным набором аргументов. Например, функция `Integrate` в пакете `Mathematica` может вычислять как неопределенный, так и определенный, и даже двойной интеграл

```
Integrate[Cos[Sqrt[x]]]
```

```
Integrate[x Log[x], {x,1,2}]
```

```
Integrate[Sqrt[x^2+y^2], {x,0,1}, {y,0,1}]
```

В зависимости от количества аргументов и их типов происходит выбор конкретной функции. Об этой особенности стоит помнить. Какая

именно из одноименных функций будет вызвана, определяется, вообще говоря, сигнатурой, то есть набором типов аргументов в списке параметров вызываемой функции. Такой подход позволяет использовать одни и те же названия функций везде, где используются одинаковые математические обозначения, и перенести все внимание на самые существенные особенности реализуемых алгоритмов. Наряду с арсеналом аналитических инструментов и богатыми библиотеками готовых алгоритмов, это позволяет ускорить решение сложных задач.

Выполнение лабораторных работ представляет собой первое знакомство с несколькими каплями в море возможностей, заложенных в пакеты прикладных математических программ.

Список литературы

1. *Говорухин, В. Н.* Компьютер в математическом исследовании: Maple, MATLAB, LaTeX / В. Н. Говорухин, В. Г. Цибулин. — СПб.: Питер, 2002. — 624 с.
2. *Эдвардс, Ч. Г.* Дифференциальные уравнения и краевые задачи: моделирование и вычисление с помощью Mathematica, Maple и MATLAB. / Ч. Г. Эдвардс, Д. Э. Пенни. — Киев.: Диалектика-Вильямс, 2007.
3. *Воробьев, Е. М.* Введение в систему «Математика» / Е. М. Воробьев. — М.: Финансы и статистика, 2001. — 264 с.
4. *Wolfram, S.* The Mathematica book. 5-th edition. [Электронный ресурс] / S. Wolfram. — 2003. — 1301 p. — www.wolfram.com.
5. *Потемкин, В. Г.* Система инженерных и научных расчетов MatLab 5.x (в 2-х томах) / В. Г. Потемкин. — М.: Диалог-МИФИ, 1999.
6. *Дьяконов, В. П.* MatLab 6. Учебный курс / В. П. Дьяконов. — СПб.: Питер, 2002. — 592 с.
7. *Поршнев, С. В.* MatLab 7. Основы работы и программирования / С. В. Поршнев. — М.: Бином-Пресс, 2006. — 320 с.
8. <http://computers.plib.ru/math>, <http://exponenta.ru>,
<http://ru.wikipedia.org/wiki/matlab>,
<http://ru.wikipedia.org/wiki/mathematica>,
<http://ru.wikipedia.org/wiki/maple>
9. *Рабинович, М. И.* Введение в теорию колебаний и волн / М. И. Рабинович, Д. И. Трубецков. — 3-е изд. — Москва–Ижевск: НИЦ «Регулярная и хаотическая динамика», 2000. — 560 с.
10. *Горяченко, В. Д.* Элементы теории колебаний: учеб. пособие для вузов / В. Д. Горяченко. — 2-е изд. — М.: Высш. шк., 2001. — 395 с.
11. *Lorenz, E. N.* Deterministic Nonperiodic Flow / E. N. Lorenz // Journal of the Atmospheric Sciences. — 1963. — Vol. 20. — P. 130–141.

12. *Хакен, Г.* Синергетика / Г. Хакен. — М.: Мир, 1980. — 404 с.
13. *Анищенко, В. С.* Сложные колебания в простых системах: Механизмы возникновения, структура и свойства динамического хаоса в радиофизических системах / В. С. Анищенко. — 2-е изд. — М.: «ЛИБРОКОМ», 2009. — 320 с.
14. *Мун, Ф.* Хаотические колебания / Ф. Мун. — М.: Мир, 1990. — 312 с.
15. *Кузнецов, С. П.* Динамический хаос / С. П. Кузнецов. — М.: Физматлит, 2001. — 298 с.
16. *Шильников, А. Л.* Бифуркации и хаос в модели Мариока–Шимицу / А. Л. Шильников // Методы качественной теории дифференциальных уравнений: Межвуз. сб. Горьк. ун-т. — Горький. — 1988. — С. 130–138.
17. *Rössler, O. E.* An equation for continuous chaos / O. E. Rössler // Phys. Lett. A — 1976. — Vol. 57, №5. — P. 397–398.
18. *May, R. M.* Nonlinear Aspects of Competition Between Two Species / R. M. May, W. J. Leonard // SIAM J. Appl. Math. — 1975. — Vol. 29, №2. — P. 243–253.
19. *Ханин, Я. И.* Основы динамики лазеров / Я. И. Ханин. — М.: Наука. Физматлит, 1999. — 368 с.
20. *Frederickson, P.* The Lyapunov dimension of strange attractors / P. Frederickson, J. Kaplan, J. Yorke // J. Diff. Equations — 1983. — Vol. 49. — P. 185–194.
21. *Малинецкий, Г. Г.* Современные проблемы нелинейной динамики / Г. Г. Малинецкий, А. Б. Потапов. — 2-е изд. — М.: Едиториал УРСС, 2002. — 360 с.
22. *Benettin, G.* Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them / G. Benettin [et al.] // Mechanica. — 1980. — Vol. 15, №1. — P. 9–30.

23. *Глызин, Д. С.* Метод динамической перенормировки для нахождения максимального ляпуновского показателя хаотического аттрактора / Д. С. Глызин, С. Д. Глызин, А. Ю. Колесов, Н. Х. Розов // Дифференциальные уравнения. — 2005. — Т. 41, №2. — С. 268–273.
24. *Андронов А. А.* Теория колебаний / А. А. Андронов, А. А. Витт, С. Э. Хайкин. — 2-е изд. — М.: Физматгиз, 1959. — 926 с.
25. *Беллман, Р.* Дифференциально-разностные уравнения / Р. Беллман, К. Л. Кук. — М.: Мир, 1967. — 548 с.
26. *Эльсгольц, Л. Э.* Введение в теорию дифференциальных уравнений с отклоняющимся аргументом / Л. Э. Эльсгольц, С. Б. Норкин. — М.: Наука, 1971. — 296 с.
27. *Красовский, Н. Н.* Некоторые задачи теории устойчивости движения / Н. Н. Красовский. — М.: Физматгиз, 1959.
28. *Глызин, С. Д.* Численные методы анализа динамических систем / С. Д. Глызин. — Ярославль: ЯрГУ, 2002. — 76 с.
29. *Колмогоров, А. Н.* Элементы теории функций и функционального анализа / А. Н. Колмогоров, С. В. Фомин. — 4-е изд. — М.: Наука, 1976.
30. *Краснов, М. Л.* Интегральные уравнения. Введение в теорию / М. Л. Краснов. — 2-е изд. — М.: КомКнига, 2006. — 304 с.
31. *Цлаф, Л. Я.* Вариационное исчисление и интегральные уравнения / Л. Я. Цлаф. — 2-е изд. — М.: Наука, 1970. — 192 с.
32. *Васильева, А. Б.* Дифференциальные и интегральные уравнения, вариационное исчисление в примерах и задачах / А. Б. Васильева, Г. Н. Медведев, Н. А. Тихонов, Т. А. Уразгильдина. — 2-е изд. — М.: Физматлит, 2005. — 432 с.
33. *Самко, С. Г.* Интегралы и производные дробного порядка и некоторые их приложения / С. Г. Самко, А. А. Килбас, И. О. Маричев. — Минск: Наука и техника, 1987. — 688 с.
34. *Заславский, Г. М.* Физика хаоса в гамильтоновых системах / Г. М. Заславский. — Москва–Ижевск: Институт компьютерных исследований, 2004. — 288 с.

Приложение

В приложении приведены некоторые функции, которые могут оказаться полезными при реализации различных алгоритмов. Обратиться при необходимости к подробному описанию той или иной функции с примерами ее использования можно, набрав в окне для ввода команд и выполнив строку

?<имя функции> (* Mathematica, например, ?Plot *)

?<имя функции> # Maple, например, ?int

help <имя функции> % MATLAB, например, help ode45

или воспользовавшись традиционным доступом к справке через F1. Из окна справки можно перенести нужный пример в основное рабочее пространство пакета и модифицировать его применительно к своей задаче. Тем самым на первых порах можно значительно снизить вероятность возникновения ошибки при создании кода. Символы (**), #, % позволяют вставлять комментарии в тексты программ в соответствующей среде. Используемая в ряде примеров команда with(linalg): пакета Maple подключает библиотеку с функциями из линейной алгебры.

Перечислим некоторые часто употребляемые функции.

Проинтегрировать функцию можно следующим образом:

Integrate (* см. примеры, приведенные в заключении *)

int # например, int(x/(x^3+1), x); или int(exp(-x)*sin(x), x=0..infinity);

quad % например, quad(@(x)1./(x.^3-2*x-5),0,2)

Отметим, что в каждом из пакетов существует целое семейство функций, позволяющих вычислять интегралы различных типов.

Графики функциональных зависимостей можно легко построить при помощи команд

Plot (* например, Plot[{Sin[x], Sin[2 x], Sin[3*x]}, {x, 0, 2 Pi}] *)

plot # plot([sin(x), x-x^3/6], x=0..2, color=[red,blue], style=[point,line]);

plot % например, x = 0:0.1:5*pi; plot(x, sin(x))

В каждом из пакетов существуют широчайшие возможности для графического представления результатов с помощью десятков различных функций.

Собственные значения матрицы вычисляются следующим образом:

Eigenvalues (* например, Eigenvalues[{a, b}, {c, d}] *)

eigenvals # например, with(linalg): eigenvals([a,b],[c,d]);

eig % например, eig([1 2; 3 4])

Определить коэффициенты характеристического многочлена матрицы можно с помощью функций
`CharacteristicPolynomial` (* например, квадратный трехчлен матрицы 2×2 относительно переменной x строится при помощи команды вида `CharacteristicPolynomial[{a, b}, {c, d}, x]` *)
`CoefficientList` (* например, `CoefficientList[2+a*x+x^3, x]` *)
`charpoly` # например, `with(linalg): charpoly([a,b],[c,d], lambda);`
`poly` % например, `poly([1 2; 3 4])`

Построить матрицу Якоби векторного поля произвольной размерности можно следующим образом:
`Jacob[f_, x_] := Outer[D[#1, #2]&, f, x];` (* поскольку стандартной функции в Mathematica нет, определяем ее самостоятельно и тестируем на примере: `Jacob[{y[t], -x[t]+y[t]*z[t], 1-y[t]^2}, {x[t], y[t], z[t]}` *)
`jacobian` # например, `with(linalg): jacobian([r*cos(phi), r*sin(phi)], [r, phi]);`
`jacobian` % например, `jacobian([x*y*z; y; x+z], [x, y, z])`

Решение систем линейных алгебраических уравнений вида $Ay=b$, где A — матрица размера $n \times n$, b — вектор, возможно с помощью таких инструментов:
`LinearSolve` (* например, `LinearSolve[{a, b}, {c, d}, {p, q}]` *)
`linsolve` # например, `with(linalg): linsolve([1,2],[3,4], [5,6]);`
 $A \backslash b$ % где \backslash операция левостороннего деления, эквивалентная $A^{-1}b$, предполагается, что переменные A и b ранее определены.

При необходимости заменить одно выражение в тексте программы другим выражением (то есть использовать правило преобразования) следует воспользоваться функциями
`ReplaceAll` (* в инфиксной форме `/.`, например, `{x, x^2, y, z} /. x->a` *)
`subs` # например, `subs(sin(x)=y, sin(x)/sqrt(1-sin(x)))`;
`subs` % например, `subs(x*y+z, x, 3)`

Особо отметим весьма широкие возможности численного решения систем дифференциальных уравнений с помощью пакетов прикладных математических программ. Для этого круга задач предназначены следующие функции.
`NDSolve` (* «Универсальный решатель» ДУ системы Mathematica *)
`NDSolve[{x'[t]==-y[t]-x[t]^2, y'[t]==x[t]-y[t]^3, x[0]==y[0]==1}, {x, y}, {t,20}];` `ParametricPlot[Evaluate[{x[t], y[t]} /. %], {t,0,20}]` (* строит фазовый портрет системы ОДУ на плоскости $(x[t], x'[t])$, символ «%» означает результат предыдущей команды *)

```
sol=NDSolve[{x'[t]==x[t]*(x[t-Pi]-x[t-1]), x[t/;t<=0]==Cos[t]}, x,{t,0,8}];
Plot[Evaluate[{x[t], x'[t]} /. First[sol]], {t,0,8}, PlotRange->All] (* парой
команд строится зависимость решения  $x(t)$  уравнения с запаздывани-
ем, а также  $x'(t)$  от времени  $t$  *)
```

```
NDSolve[{D[u[t,x],t]==D[u[t,x],x,x], u[0,x]==0, u[t,0]==Sin[t], u[t,5]==0},
u, {t,0,10}, {x,0,5}]; (* численно решает уравнение теплопроводности *)
Plot3D[Evaluate[u[t,x]/.%], {t,0,10}, {x,0,5}, PlotRange->All]
(* строит график решения u(t,x) уравнения теплопроводности *)
```

deplot # функция deplot библиотеки detools пакета Maple численными методами находит решение ДУ и строит его график

with(DEtools):

```
DEplot([diff(x(t),t)=x(t)*(1-y(t)), diff(y(t),t)=.3*y(t)*(x(t)-1)], [x(t), y(t)],
t=-7..7, [[x(0)=1.2, y(0)=1.2], [x(0)=1, y(0)=.7]], stepsize=.2);
```

численно решает систему ОДУ и строит ее фазовый портрет

ode23, ode45 % две функции большого семейства алгоритмов MATLAB для решения ОДУ

dde23 % служит для численного решения систем с запаздыванием

% Пример скрипта, записанного в файле ddex11.m, для решения дифференциального уравнения с двумя запаздываниями:

```
function ddex11 % решаем уравнение  $y' = y * [y(t-\pi) - y(t-1)]$ 
```

```
sol = dde23(@ddex11de, [1,pi], @ddex11hist, [0,8]);
```

```
figure; plot(sol.x, sol.y)
```

```
title('DDE example');
```

```
xlabel('time t'); ylabel('solution y');
```

```
function s = ddex11hist(t)
```

```
s = cos(t); % начальная функция
```

```
function dydt = ddex11de(t,y,Z) % задаем уравнение
```

```
ylag1 = Z(:,1); ylag2 = Z(:,2);
```

```
dydt = [y(1)*(ylag2(1)-ylag1(1))];
```

Учебное издание

Глазков Дмитрий Владимирович

**Пакеты прикладных
математических программ**

*Методические указания
к проведению лабораторных работ*

Редактор, корректор И. В. Бунакова
Компьютерная верстка Д. В. Глазков

Подписано в печать 28.09.2009. Формат 60х84/16.
Бумага офсетная. Гарнитура "Times New Roman".

Усл. печ. л. 2,32. Уч.-изд. л. 1,5.

Тираж 100 экз. Заказ .

Оригинал-макет подготовлен
в редакционно-издательском отделе
Ярославского государственного университета
им. П. Г. Демидова.

Отпечатано на ризографе.

Ярославский государственный университет
им. П. Г. Демидова.

150000, Ярославль, ул. Советская, 14.